

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Estimador de Consumo de Combustível para Dispositivos de Computação Móvel

Eduardo João Silva Varão Moreira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Doutor Rui Esteves Araújo

30 de Junho de 2014

A Dissertação intitulada

“Estimador de Consumo de Combustível para Dispositivos de Computação Móvel”

foi aprovada em provas realizadas em 25-07-2014

o júri

Presidente


Professor Doutor Adriano da Silva Carvalho

Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Alberto Jorge Lebre Cardoso

Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Ciências da Universidade de Coimbra


Professor Doutor Rui Manuel Esteves Araújo

Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.


Autor - Eduardo João Silva Varão Moreira

Faculdade de Engenharia da Universidade do Porto

Resumo

A conceção de um estimador de consumo de combustível para dispositivos de computação móvel emerge de considerações de ordem tecnológica, associadas a preocupações de cariz económico e ambiental. Parte-se da ideia de que, numa sociedade em que o veículo convencional se assume como um importante meio de deslocação das pessoas, importa encontrar ferramentas que atenuem o impacto deste no consumo dos recursos naturais e nos custos económicos, nomeadamente dos utilizadores. Assim, na presente dissertação, estudam-se diferentes modelos físicos e empíricos que modelam o consumo de combustível de um automóvel, incidindo em maior profundidade sobre um modelo físico cujas entradas são a velocidade, a aceleração e a inclinação, dados que podem ser obtidos através dos sensores dos *smartphones*. Baseando-se neste modelo, desenvolveu-se um algoritmo para estimativa de consumo num dispositivo móvel.

Na primeira fase, tendo por base a ferramenta Advisor, analisou-se o modelo e realizou-se uma calibração, recorrendo a vários ciclos de condução, constantes e variáveis. Subsequentemente, realizou-se uma análise à sensibilidade do modelo, verificando-se de que forma a variação das entradas provocaria mudanças na saída do estimador. Esta análise foi complementada com a verificação da influência da variação do período de amostragem e da introdução de ruído, bem como do método utilizado para obter a aceleração como derivada da velocidade, tendo-se concluído que o modelo poderia ser transformado num estimador de consumo de combustível para um dispositivo móvel, principalmente pelas suas características de robustez e facilidade de implementação.

Na segunda fase, constatou-se que o modelo, apesar de ser bastante robusto, não conseguia acompanhar o consumo de forma correta para todos os tipos de ciclos. Para controlar este problema, optou-se por criar um algoritmo de deteção de ciclo, cuja saída deveria ser utilizada na estimativa do consumo final. Este recorre aos últimos 50 segundos para obter a velocidade média, o tempo parado e a energia de aceleração média, a partir dos quais, com base num algoritmo de lógica difusa, identifica o tipo de ciclo que o veículo se encontra a percorrer. Seguidamente, modificou-se o modelo usado na primeira fase, considerando que a inclinação seria sempre zero, eliminando a segunda eficiência e passando a outra eficiência a variável, ou seja, esta teria um termo cujo valor seria fixo e possuiria outro termo, este dependente da saída do algoritmo de deteção de ciclo de condução e de uma constante associada a cada tipo de ciclo.

Na última fase de desenvolvimento, implementaram-se os dois algoritmos num dispositivo Android, permitindo, desta forma, verificar, em tempo real, qual o consumo de combustível do veículo. Paralelamente a esta implementação, integrou-se um sistema de aquisição que permite guardar todos os dados adquiridos, tornando possível analisá-los posteriormente.

Os resultados encontrados nos vários testes/ciclos efetuados foram satisfatórios, sobretudo, quando comparados com os resultados provenientes da plataforma Advisor. A aplicação desenvolvida revelou-se, portanto, como uma possível ferramenta de monitorização de consumo de combustível, pelo que seria de dar sequência ao estudo desenvolvido, analisando a possibilidade de implementar e testar este algoritmo para diferentes veículos.

Abstract

The design of an estimator of fuel consumption for mobile computing devices emerging from technological considerations associated with concerns for economic and environmental nature. Part is the idea that, in a society where the conventional vehicle is assumed to be an important means of displacement of persons, it find tools that mitigate the impact of the consumption of natural resources and economic costs, particularly for users. Thus, in this dissertation, we study different physical and empirical models that model the consumption of an automobile fuel, focusing in greater depth on a physical model whose inputs are the speed, acceleration and inclination data that can be obtained through sensors of smartphones. Based on this model, we developed an algorithm to estimate consumption on a mobile device.

In the first stage, based on the Advisor tool, we analyzed the model and carried out a calibration, using various driving cycles, constants and variables. Subsequently, we performed a sensitivity analysis on the model, verifying how the variation of inputs cause changes in the output of the estimator. This analysis was supplemented by checking the influence of the variation of the sampling period and the introduction of noise as well as the method used to obtain the acceleration as a derivative of the velocity, having concluded that the model could be transformed into an estimator consumption fuel for a mobile device, mainly due to its characteristics of robustness and ease of implementation.

In the second phase, it was found that the model, despite being quite robust, could not follow the consumption of correct form for all types of cycles. To control this problem, we chose to create a cycle detection algorithm, whose output should be used in the estimation of final consumption. This refers to the last 50 seconds to obtain the average speed, the stop time and the average acceleration energy, from which, based on a fuzzy logic algorithm identifies the type of cycle which the vehicle is to travel. Subsequently modified the model used in the first phase, whereas the slope would always be zero, eliminating the second passing efficiency and other efficiency variable, ie, this would have a term whose value would be fixed and would own another term, this dependent the detection output of the driving cycle and a constant associated with each type of cycle algorithm.

In the last phase of development, were implemented both algorithms on an Android device, allowing thus verify, in real time, what the fuel consumption of the vehicle. Alongside this implementation, was integrated into an acquisition system that allows you to save all acquired data, making it possible to analyze them further.

The results obtained in the various tests / cycles were performed satisfactory, especially when compared with the results from the Advisor platform. Therefore, the developed application is revealed as a possible tool for monitoring fuel consumption, so it would be to follow up to the study, analyzing the possibility to implement and test this algorithm for different vehicles.

Agradecimentos

Agradeço a todos os professores que me acompanharam e incentivaram em todo o meu percurso académico. Particularmente, ao professor Rui Esteves Araújo expresse o meu profundo agradecimento pela orientação, motivação e apoio incondicional, que muito estimularam o meu desejo de querer, sempre, saber mais e a vontade constante de fazer melhor.

À minha família, sobretudo aos meus pais e irmãs, pelo apoio para que concluísse todas as etapas da minha vida, mostrando-me as minhas capacidades e apoiando todas as minhas decisões. Espero que, com a conclusão desta etapa, consiga retribuir todo o carinho, apoio e dedicação que, constantemente, me oferecem.

À Rita, um agradecimento especial pela dedicação, pela entrega, pela ajuda e por todos os incentivos diários, que muito me motivaram para a conclusão desta tese.

Por fim, agradeço a todos os meus amigos, nomeadamente ao André Serra, ao António Franco, ao Carlos Ferreira, ao João Oliveira, ao Joaquim Figueiredo, ao Luís Hermenegildo, ao Nuno Velho, ao Tiago Cascão e ao Tiago Reis, pela amizade, companheirismo e ajuda durante este meu percurso académico, fatores muito importantes na realização desta tese.

Eduardo Moreira

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Enquadramento	2
1.3	Objetivos da Dissertação	3
2	Revisão Bibliográfica	5
2.1	Tipos de Modelos	5
2.1.1	Modelos Mentais / Intuitivos	5
2.1.2	Modelos Matemáticos	5
2.1.3	Modelos Gráficos	6
2.2	Modelos Matemáticos para Estimativa de Consumo de Combustível	6
2.2.1	Modelos Físicos	7
2.2.2	Modelos Empíricos	13
2.3	Sistema de Aquisição de Dados Vocacionados para Automóveis	17
3	Análise dos Modelos Matemáticos	19
3.1	Advisor: Ferramenta para Validação dos Modelos	19
3.1.1	Principais Blocos do Modelo BD_CONV	19
3.1.2	Interface	30
3.2	Análise e Calibração de modelos	33
3.2.1	Seleção de Modelos para Estimativa	33
3.2.2	Calibração dos Modelos	34
3.2.3	Resultados da Calibração	37
3.3	Sensibilidade do Modelo Físico	38
3.3.1	Sensibilidade em Função da Aceleração	39
3.3.2	Sensibilidade em Função da Velocidade	42
3.3.3	Sensibilidade em Função da Inclinação	45
3.3.4	Conclusões	49
3.4	Simulação do Modelo Físico	49
3.4.1	Efeitos de Diferentes Períodos de Amostragem	49
3.4.2	Impacto do Ruído de Aquisição nas Estimativas	54
3.4.3	Estudo das Influências dos Métodos de Cálculo da Aceleração	58
3.5	Conclusões	61
4	Formulação do Novo Algoritmo	63
4.1	Algoritmo para Detecção de Tipo de Ciclo de Condução	63
4.1.1	Entradas e Saídas do Algoritmo	63
4.1.2	Funções de Pertença e Regras de Inferência	64

4.1.3	Testes do Algoritmo	66
4.2	Algoritmo de Estimativa de Consumo	72
4.2.1	Equações Matemáticas do Estimador	72
4.2.2	Resultados do Estimador	74
4.3	Conclusões	78
5	Implementação e Teste do Algoritmo em Sistema Operativo Android	79
5.1	Aquisição de Entradas	79
5.1.1	Variáveis a adquirir	79
5.1.2	Método de gravação de dados	80
5.1.3	Tamanhos exemplificativos dos ficheiros	81
5.2	Interface gráfica	81
5.3	Algoritmo implementado	82
5.3.1	Resultados da aquisição	83
5.3.2	Resultados da identificação de ciclo	89
5.3.3	Resultados da estimativa do consumo	90
5.4	Conclusões	97
6	Conclusão	99
6.1	Principais Resultados	99
6.2	Contribuições	100
6.3	Trabalho futuro	100
A	Anexos do modelo simulado	101
A.1	Diagrama do algoritmo de identificação de tipo de ciclo	101
A.2	Diagrama do modelo implementado	103
B	Anexos do código implementado	105
B.1	Código da atividade principal	105
B.2	Código das classes dos estimador	117
B.2.1	Classe Car	117
B.2.2	Classe CarEnvironment	120
B.2.3	Classe VehicleState	121
B.2.4	Classe Position	124
B.2.5	Classe Estimator	125
	Referências	129

Lista de Figuras

3.1	Visão Geral do Modelo BD_CONV[27]	20
3.2	Bloco vhe[27]	21
3.3	Bloco wh[27]	22
3.4	Bloco axle[27]	23
3.5	Bloco fd[27]	24
3.6	Bloco gb[27]	25
3.7	Bloco cl[27]	26
3.8	Bloco acc[27]	27
3.9	Bloco fc[27]	28
3.10	Bloco ex[27]	29
3.11	Bloco vc[27]	30
3.12	Interface Advisor - Definição de veículo[27]	31
3.13	Interface Advisor - Definição de ciclo de condução[27]	32
3.14	Interface Advisor - Apresentação de resultados[27]	33
3.15	Exemplo de consumo dos modelos 2 e 13	36
3.16	Ciclo de calibração 1	37
3.17	Ciclo de calibração 2	37
3.18	Ciclo de calibração 3	38
3.19	Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 6^\circ$	39
3.20	Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 3^\circ$	40
3.21	Modelo 2 - Sensibilidade do consumo em função de a com $\theta = -6^\circ$	40
3.22	Modelo 2 - Sensibilidade do consumo em função de a com $\theta = -3^\circ$	41
3.23	Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 0^\circ$	41
3.24	Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 6^\circ$	43
3.25	Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 3^\circ$	43
3.26	Modelo 2 - Sensibilidade do consumo em função de v com $\theta = -6^\circ$	44
3.27	Modelo 2 - Sensibilidade do consumo em função de v com $\theta = -3^\circ$	44
3.28	Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 0^\circ$	45
3.29	Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 6^\circ$	46
3.30	Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 3^\circ$	47
3.31	Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = -6^\circ$	47
3.32	Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = -3^\circ$	48
3.33	Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 0^\circ$	48
3.34	Comparação de consumo com diferentes Ts para o ciclo: CYC CONSTANT	50
3.35	Comparação de consumo com diferentes Ts para o ciclo: CYC CONSTANT 60	51
3.36	Comparação de consumo com diferentes Ts para o ciclo: CYC CLEVELAND	52
3.37	Comparação de consumo com diferentes Ts para o ciclo: CYC HWFET	52
3.38	Comparação de consumo com diferentes Ts para o ciclo: CYC MANHATTAN	53

3.39	Comparação de consumo com diferentes Ts para o ciclo: CYC UDDS	54
3.40	Comparação de consumo, introduzindo ruído para o ciclo: CYC CONSTANT . . .	55
3.41	Comparação de consumo, introduzindo ruído para o ciclo: CYC CONSTANT 60 . .	55
3.42	Comparação de consumo, introduzindo ruído para o ciclo: CYC CLEVELAND . .	56
3.43	Comparação de consumo, introduzindo ruído para o ciclo: CYC HWFET	57
3.44	Comparação de consumo, introduzindo ruído para o ciclo: CYC MANHATTAN . .	57
3.45	Comparação de consumo, introduzindo ruído para o ciclo: CYC UDDS	58
3.46	Exemplo de derivação numérica 1	60
3.47	Exemplo de derivação numérica 2	60
3.48	Exemplo de derivação numérica 3.1	61
3.49	Exemplo de derivação numérica 3.2	61
4.1	Função de pertença da velocidade média	64
4.2	Função de pertença do bloco de lógica difusa para o tempo parado	65
4.3	Função de pertença do bloco de lógica difusa para Energia de Aceleração	65
4.4	Função de pertença do bloco de lógica difusa para a saída	65
4.5	Teste de identificação de ciclo para CYC MANHATTAN	67
4.6	Teste de identificação de ciclo para CYC UDDS	68
4.7	Teste de identificação de ciclo para CYC HWFET	69
4.8	Teste de identificação de ciclo para CYC CLEVELAND 90	70
4.9	Teste de identificação de ciclo para CYC CLEVELAND	71
4.10	Teste de identificação de ciclo para CYC HWFET 110	72
4.11	Resultados do novo algoritmo para o ciclo MANHATTAN	74
4.12	Resultados do novo algoritmo para o ciclo UDDS	75
4.13	Resultados do novo algoritmo para o ciclo HWFET	75
4.14	Resultados do novo algoritmo para o ciclo COMBINADO	76
4.15	Resultados do novo algoritmo para o ciclo CLEVELAND	77
4.16	Resultados do novo algoritmo para o ciclo Barcelos Maia	77
5.1	Ficheiro exemplificativo do sistema de aquisição	80
5.2	Relação entre tamanho do ficheiro e tempo de aquisição	81
5.3	Interface Android	82
5.4	Ciclo obtido a 65 km/h	84
5.5	Ciclo obtido a 70 km/h	85
5.6	Ciclo obtido a 75 km/h	86
5.7	Ciclo obtido a 100 km/h	86
5.8	Ciclo entre Maia e FEUP	87
5.9	Ciclo entre Maia e FEUP - Localização GPS	88
5.10	Ciclo entre FEUP e Maia	88
5.11	Ciclo Urbano	89
5.12	Ciclo entre Foz e Moreira	90
5.13	Apresentação do resultado final para ciclo 1	92
5.14	Apresentação do resultado final para ciclo 10	93
5.15	Apresentação do resultado final para ciclo 14	94
5.16	Apresentação do resultado final para ciclo 35	95
5.17	Apresentação do resultado final total	96
A.1	Diagrama do modelo de identificação de tipo de ciclo	102
A.2	Diagrama do modelo implementado	104

Lista de Tabelas

2.1	Saídas do modelo 1	7
2.2	Entradas do modelo 1	7
2.3	Parâmetros do modelo 1	8
2.4	Saídas do modelo 2	8
2.5	Entradas do modelo 2	9
2.6	Parâmetros do modelo 2	9
2.7	Saídas do modelo 3	9
2.8	Entradas do modelo 3	9
2.9	Parâmetros do modelo 3	10
2.10	Saídas do modelo 4	10
2.11	Entradas do modelo 4	10
2.12	Parâmetros do modelo 4	10
2.13	Saídas do modelo 5	11
2.14	Entradas do modelo 5	11
2.15	Parâmetros do modelo 5	11
2.16	Saídas do modelo 6	12
2.17	Entradas do modelo 6	12
2.18	Parâmetros do modelo 6	12
2.19	Saídas do modelo 7	12
2.20	Entradas do modelo 7	13
2.21	Parâmetros do modelo 7	13
2.22	Saídas do modelo 8	13
2.23	Entradas do modelo 8	14
2.24	Parâmetros do modelo 8	14
2.25	Saídas do modelo 9	14
2.26	Entradas do modelo 9	14
2.27	Parâmetros do modelo 9	15
2.28	Saídas do modelo 10 e 11	15
2.29	Entradas do modelo 10 e 11	15
2.30	Parâmetros do modelo 10 e 11	16
2.31	Saídas do modelo 12	16
2.32	Entradas do modelo 12	16
2.33	Parâmetros do modelo 12	16
2.34	Saídas do modelo 13	17
2.35	Entradas do modelo 13	17
2.36	Parâmetros do modelo 13	17
3.1	Comparação entre modelos físicos	34

3.2	Comparação entre modelos empíricos	34
3.3	Parâmetros obtidos diretamente do Advisor	35
3.4	Parâmetros obtidos indiretamente do Advisor	35
3.5	Massas obtidas através de aproximação	35
3.6	Parâmetros obtidos através de calibração	36
3.7	Valores da sensibilidade de a para valores típicos de a e v	42
3.8	Valores da sensibilidade de v para valores típicos de a e v	45
3.9	Valores da sensibilidade de v para valores típicos de a e v	49
3.10	Coeficientes de derivadas de primeira ordem	59
4.1	Regras usando velocidade média e aceleração	66
4.2	Regras usando velocidade média e tempo parado	66
4.3	Escala para identificação de ciclo de condução	66
4.4	Saídas do modelo final	73
4.5	Entradas do modelo final	73
4.6	Parâmetros do modelo 2	73
5.1	Tamanhos exemplificativos dos ficheiros	81
5.2	Parâmetros obtidos para o Alfa Romeo MITO	91
5.3	Parâmetros estimados para o Alfa Romeo MITO	91
5.4	Consumos e erros da estimativa de consumo	97

Abreviaturas e Símbolos

Advisor	Advanced Vehicle Simulator
BD_CONV	Modelo simulink de um veículo convencional
vhe	Veículo
wh	Roda e eixo dianteiro/traseiro
axle	Conversor de eixos para linha de transmissão
fd	Transmissão final
gb	Caixa de velocidades
cl	Embreagem
acc	Cargas de Acessórios Mecânicos
fc	Conversor de Combustível
ex	Sistema de Escape
vc	Sistema de controlo do automóvel
cyc	Ciclo de condução
sdo	Sistema de interface
Ts	Período de Amostragem
a	Aceleração
v	Velocidade

Capítulo 1

Introdução

A presente dissertação, elaborada no âmbito do Mestrado Integrado em Engenharia Electrotécnica e de Computadores (MIEEC) da Faculdade de Engenharia da Universidade do Porto, tem por finalidade apresentar todo o trabalho realizado no desenvolvimento de um estimador de consumo de combustível para dispositivos de computação móvel. Inicia-se com a revisão da bibliografia, relacionada com a temática em estudo, bem como com o trabalho de pesquisa sobre ferramentas passíveis de serem utilizadas durante a realização da mesma. Foca-se, depois, mais profundamente num dos modelos capazes de estimarem corretamente o consumo de combustível, explicitando, por último, o algoritmo implementado, assim como todos os métodos considerados para a sua conceção. Assim, organiza-se em seis capítulos, a saber:

- Capítulo 1 (Introdução): neste capítulo é refere-se a motivação para a sua realização, e repetivo enquadramento. São, ainda, apresentados os objetivos da dissertação.
- Capítulo 2 (Revisão Bibliográfica): neste capítulo expõe-se, de forma detalhada, o levantamento bibliográfico relacionado com a temática da dissertação.
- Capítulo 3 (Análise de modelos de consumo de combustível): neste capítulo são abordados os modelos mais relevantes para a dissertação, bem como os resultados das suas simulações *offline*.
- Capítulo 4 (Formulação do novo Algoritmo): neste capítulo será descrito um novo algoritmo desenvolvido para cálculo dos consumos, baseado no estudado no capítulo 3, e um algoritmo de deteção de ciclo de condução.
- Capítulo 5 (Implementação do algoritmo em Android): neste capítulo será apresentada a implementação em Android do algoritmo desenvolvido, bem como os resultados, intermédios e finais, desta implementação.
- Capítulo 6 (Conclusões): neste capítulo enunciam-se as conclusões do trabalho, os principais resultados, respetivas contribuições, terminando-se com hipóteses de trabalhos futuros.

1.1 Motivação

No mundo atual, grande parte da população desloca-se, nas suas atividades diárias, usando veículo convencional[1]. Esta elevada utilização de veículos terá como consequência direta o aumento do consumo de combustíveis, facto que poderá ter um duplo impacto negativo. Não só poderá ter consequências negativas do ponto de vista ambiental, devido aos gases poluentes emitidos e à diminuição das reservas de petróleo, como também, ao nível económico, poderá provocar o aumento dos preços dos mesmos [1, 2]. Torna-se, assim, fundamental encontrar formas de racionalizar a utilização de combustível, quer no que se refere aos níveis de dependência dos produtos petrolíferos, quer no que se refere aos níveis de consumo. Relativamente à redução dos níveis de consumo, os métodos que têm vindo a ser utilizados para o efeito incluem uso de veículos com bons índices de consumo, uso de rotas mais eficientes (Menos tráfego e velocidades constantes) e condução defensiva ou *Eco-Driving*[3]. Quanto ao primeiro método, é perceptível que este não se encontra ao alcance de todos os utilizadores, uma vez que os veículos com melhores índices são, não raramente, os veículos de custo mais elevado. Por sua vez, o segundo método implica a existência de alternativas para a rota a tomar, o que nem sempre acontece. Por fim, o terceiro pode ser considerado uma estratégia emergente, permitindo poupar combustível de forma instantânea. Este é já apoiado por fabricantes de automóveis que introduziram nos veículos dispositivos de monitorização instantânea de consumos, com o objetivo de induzir no condutor hábitos e práticas de boa condução. Igualmente, o desenvolvimento e disseminação dos equipamentos móveis de utilização pessoal, tais como *smartphones*[4, 5] capazes de elevado desempenho computacional e com diferentes tipos de sensores integrados[6], tornam possível observar o aparecimento de novas aplicações, permitindo uma utilização destes aparelhos para além do simples uso como telefone [5]. Mais especificamente, possibilitam o uso dos seus sensores para observar o estado do veículo e, assim, compreender formas de poupança de energia. Conseguir-se-ia, deste modo, uma monitorização dos consumos de combustível, mesmo nos veículos não equipados com mecanismos de monitorização instantânea de combustível. Acresce que, a possibilidade de registo de dados de consumo no dispositivo de computação móvel poderá, posteriormente, ser utilizada pelo condutor para analisar os seus consumos e tomar medidas preventivas quanto ao seu estilo de condução[7].

Partindo destas constatações, o trabalho a realizar perspetiva-se como um possível contributo para o desenvolvimento dos processos de economia de combustível, passíveis de serem utilizados por um grupo mais alargado de condutores.

1.2 Enquadramento

Neste projeto, pretende-se usar a sensorização e a capacidade de processamento dos *smartphones* para monitorização do consumo de combustível dos automóveis.

A revisão da literatura, científica, cujo foco de estudo se debruça sobre a modelação do consumo de combustível, permite verificar a existência de uma panóplia alargada de pesquisa neste

âmbito. Efetivamente, o levantamento bibliográfico demonstrou a existência de um conjunto extenso de artigos na área. Entre outros, descrevem-se modelos empíricos (baseados na percepção) e modelos físicos do automóvel, os quais possuem diversas variáveis a serem manipuladas, sendo de elevada importância as suas entradas.

No domínio deste trabalho, o enfoque recai, somente, sobre os modelos cuja informação do estado do automóvel considera unicamente a velocidade, a aceleração e a inclinação do mesmo[6, 8]. Por outro lado, será necessário conhecer algumas características do automóvel e do meio em que circula, algumas das mais relevantes serão: a massa do automóvel, a sua área frontal, o coeficiente de arrasto de ar e o coeficiente de perdas de rolamento.

1.3 Objetivos da Dissertação

Com esta dissertação visa-se criar um estimador de estado, que, com base em informação limitada, permita indicar uma estimativa do consumo instantâneo de combustível e, consequentemente, sensibilizar o condutor para a economia do mesmo. Mais especificamente, pretende-se desenvolver um algoritmo para estimação de consumo de combustível. Este será desenvolvido para que possa ser implementado num *smartphone*.

Para a concretização deste objetivo final, consideraram-se vários objetivos, correspondentes, cada um, às diferentes etapas necessárias para chegar ao desenvolvimento do algoritmo (aplicação), a saber:

- Sistematizar os modelos presentes na literatura
- Modelizar o consumo de combustível, tendo como entradas dados obtidos através dos sensores de um *smartphone* e desenvolvimento de ferramenta para validação do algoritmo
- Validar e codificar o modelo obtido
- Validar o algoritmo em ambiente real

Como objetivos pessoais pretende-se:

- Adquirir e aprofundar conhecimentos na área da programação de *smartphones*
- Desenvolver capacidades e competências de conceção e elaboração de projetos inovadores em Engenharia

Capítulo 2

Revisão Bibliográfica

Neste capítulo, sistematiza-se a informação científica recolhida no âmbito da temática abordada, ou seja, consumo de combustível e respetivos modelos de estimação, bem como temáticas relacionadas, tais como Eco-Routing e Eco-Driving. Esta serviu de referencial para o estudo empírico levado a efeito.

2.1 Tipos de Modelos

Considerando o âmbito da dissertação, tornou-se fundamental estudar os tipos de modelos de sistema usados na literatura, para tal analisou-se o capítulo 1, "Modelação paramétrica de sistemas", de [9]. O resultado desta análise é apresentado de forma sucinta nesta secção.

Antes de mais, importa compreender alguns conceitos fundamentais, designadamente o de sistema e o de modelo. Um sistema trata-se de um conjunto de elementos interligados que transformam entradas em saídas. Os sistemas têm, normalmente, um objetivo e podem ser classificados em vários tipos, de acordo com determinadas características. Um modelo é uma representação abstrata de um sistema, que se pretende que o represente da forma mais realista possível. Existem vários tipos de modelos. Neste documento serão abordados três tipos: modelos mentais/intuitivos, modelos gráficos e modelos matemáticos.

2.1.1 Modelos Mentais / Intuitivos

Os modelos mentais/intuitivos referem-se aos modelos baseados na nossa perceção da realidade. Este tipo de modelos é usado quando, por exemplo, se escreve um texto: é necessário pegar numa caneta e seguidamente realizar movimentos em determinada ordem para escrever o texto.

2.1.2 Modelos Matemáticos

Este tipo de modelos é usado quando é necessário representar uma maior complexidade do sistema e no qual existe uma descrição analítica da relação entre variáveis. Estes modelos são os

mais interessantes nesta área pelo que se realiza uma pequena análise quanto às seguintes características:

- Entradas e saídas: existem sistemas com múltiplas entradas e saídas, bem como sistemas com uma única entrada e várias saídas, o inverso e sistemas com apenas uma entrada e uma saída e são denominados, respetivamente por: MIMO, SIMO, MISO e SISO
- Lineares e não lineares: no primeiro caso as saídas dependem, apenas, de combinações lineares das entradas, enquanto que, no segundo tal não acontece.
- Paramétricos e não paramétricos: os primeiros são descritos por um conjunto de parâmetros, por seu lado, os segundos usam funções ou gráficos.
- Variantes e invariantes no tempo: os primeiros são modelos que dependem do instante em que se está a analisar o sistema, enquanto que, os segundos são invariantes de quando o sistema se inicia.
- Domínio das frequências ou dos tempos: no primeiro caso o sistema é representado com base na sua resposta em frequência, no segundo é representado de acordo com a sua evolução temporal.
- Domínio contínuo ou discreto: no primeiro caso as entradas e saídas estão relacionadas para todo o tempo, enquanto que, no segundo caso apenas estão relacionados nos tempos de amostragem.
- Determinísticos ou estocásticos: no primeiro caso as saídas podem ser obtidas mal se conhece a entrada. No segundo caso têm de ser consideradas possíveis perturbações.

Podem-se, ainda, dividir em sistemas estáticos e dinâmicos. No primeiro, caso não exista entrada também não existirá saída. O mesmo não acontece em sistemas dinâmicos.

2.1.3 Modelos Gráficos

Estes modelos definem-se por serem representações gráficos e/ou tabelas de relações entre entradas e saídas. São exemplos destes modelos os diagramas de Bode e os de Nyquist da resposta impulsional. Pode-se dizer que estes representam os modelos matemáticos de uma forma gráfica e permitem conhecer melhor a relação entre as variáveis.

2.2 Modelos Matemáticos para Estimativa de Consumo de Combustível

Os modelos apresentados neste capítulo foram escolhidos após uma primeira selecção de modelos da literatura, a qual pretendeu ser o mais exaustiva possível, sendo que as principais referências usadas foram [10, 11].

Na revisão que será exposta nesta seção, os modelos são classificados em físicos e empíricos, sendo a diferença entre classes a sua parametrização. Assim, nos modelos físicos, os parâmetros envolvidos estão diretamente relacionados com características reais do veículo, enquanto que nos modelos empíricos os parâmetros não refletem diretamente essas características. Estes serão estimados com base em testes experimentais.

2.2.1 Modelos Físicos

Esta seção é dedicada à apresentação dos modelos físicos considerados mais relevantes, bem como às referências bibliográficas onde se poderão encontrar informações mais detalhadas. Será, ainda, discutida a forma como as diversas variáveis dos modelos podem ser obtidas.

2.2.1.1 Modelo 1

$$F = \frac{1}{H_d} \left(\frac{f_{mep} N V_d}{\eta_i 2000} + \frac{P}{\eta_i} \right) \quad (2.1)$$

$$P = \frac{P_{trac}}{\eta_t} + P_{acc} \quad (2.2)$$

$$P_{trac} = mgC_r v + \frac{AC_d \rho v^3}{2} + (m + m_j)v(a + g \sin \theta) \quad (2.3)$$

Tabela 2.1: Saídas do modelo 1

Saída	Descrição	Unidades
F	Taxa de consumo	g/s
P	Potência de tração do veículo e potência dos acessórios	kW
P_{trac}	Potência de tração do veículo	kW

Tabela 2.2: Entradas do modelo 1

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
N	Velocidade do motor	rps	Rede CAN do veículo	-
P_{acc}	Potência dos acessórios	kW	Sensor específico	-
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro
a	Aceleração do veículo	m/s^2	Acelerómetro	GPS
θ	Inclinação da estrada	rad	Acelerómetro	GPS

Tabela 2.3: Parâmetros do modelo 1

Parâmetro	Descrição	Unidades
f_{mep}	Pressão efetiva média	kPa
ρ	Densidade do ar	kg/m^3
H_d	Densidade energética do diesel	kJ/kg
A	Área frontal do veículo	m^2
V_d	Deslocamento do motor	dm^3
m	Massa do veículo	kg
m_j	Massa equivalente de a inércia e das partes em movimento	kg
g	Aceleração gravítica	m/s^2
η_i	Rendimento nominal do motor	-
C_r	Coefficiente de perda de rolamento	-
C_d	Coefficiente de arrasto de ar	-

Este modelo, que é apresentado em [10, 12], está baseado em quatro parâmetros: atrito e eficiência do motor; binário máximo; transmissão e carga do veículo. Trata-se de um modelo bastante detalhado. No entanto, devido a este detalhe, os parâmetros tornam-se muito difíceis de obter e determinadas entradas exigem sensorização específica. Assim, apesar de considerar vários fatores no consumo do automóvel, torna-se muito difícil de implementar no âmbito do projeto a realizar.

2.2.1.2 Modelo 2

$$F_w = (m + m_j)a + \frac{1}{2}C_d A \rho v^2 + mg C_r \cos \theta + mg \sin \theta \quad (2.4)$$

$$f_t = f_{t_{idle}} + \frac{\delta}{\eta_i \varepsilon H_g} (mg(C_r \cos \theta + \sin \theta) + (m + m_j)av + \frac{1}{2}C_d A \rho v^3) \quad (2.5)$$

$$\delta = \begin{cases} 1 & F_w > 0 \\ 0 & F_w \leq 0 \end{cases} \quad (2.6)$$

Tabela 2.4: Saídas do modelo 2

Saída	Descrição	Unidades
F_w	Resistência total	N
f_t	Consumo instantâneo	cm^3/s
δ	Controlo de consumo quando veículo parado	-

Tabela 2.5: Entradas do modelo 2

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
θ	Inclinação da estrada	rad	Acelerómetro	GPS
a	Aceleração do veículo	m/s^2	Acelerómetro	GPS
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro

Tabela 2.6: Parâmetros do modelo 2

Parâmetro	Descrição	Unidades
g	Aceleração gravítica	m/s^2
m	Massa do veículo	kg
m_j	Massa equivalente de inércia e das partes em movimento	kg
H_g	Calor equivalente do combustível	J/cm^3
f_{idle}	Taxa de consumo em ponto morto	cm^3/s
ρ	Densidade do ar	kg/m^3
ε	Eficiência térmica do travão do motor	-
A	Área frontal do veículo	m^2
η_t	Eficiência da transmissão	-
C_d	Coefficiente de arrasto de ar	-
C_r	Coefficiente de perda de rolamento	-

O principal propósito deste modelo é o de avaliar as emissões do veículo[10, 13]. Trata-se de um modelo menos detalhado em relação ao anterior. O número e tipo de entradas deste modelo torna-o bastante interessante, isto porque são poucas e de fácil acesso. Apesar destas vantagens, o maior problema deste modelo será a calibração dos parâmetros do automóvel.

2.2.1.3 Modelo 3

$$FC = 1000\beta \frac{P_{tot}}{v} \quad (2.7)$$

Tabela 2.7: Saídas do modelo 3

Saída	Descrição	Unidades
FC	Consumo de combustível	mL/km

Tabela 2.8: Entradas do modelo 3

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
P_{tot}	Potência total requerida	kW	Sensor específico	-
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro

Tabela 2.9: Parâmetros do modelo 3

Parâmetro	Descrição	Unidades
β	Eficiência do combustível	ml/kW/s

Trata-se de um modelo apresentado em [11, 14], de origem Sul Africana. Subentende que o consumo de combustível é proporcional aos requisitos totais de energia. Com base na única equação que este modelo possui, pode-se concluir que é um modelo bastante limitado, sendo que possui ainda uma entrada muito problemática para ser obtida (P_{tot}).

2.2.1.4 Modelo 4

$$IFC = \alpha + \beta P_{tr} + \frac{\beta_2 M a^2 v}{1000} \quad (2.8)$$

Tabela 2.10: Saídas do modelo 4

Saída	Descrição	Unidades
IFC	Consumo de combustível	mL/km

Tabela 2.11: Entradas do modelo 4

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
P_{tr}	Potência de tração	kW	Sensor específico	-
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro
a	Aceleração do veículo	m/s ²	Acelerómetro	GPS

Tabela 2.12: Parâmetros do modelo 4

Parâmetro	Descrição	Unidades
β_2	Eficiência do combustível durante a aceleração	mL/(KJm/s ²)
α	Consumo de combustível em regime estacionário	mL/s
β	Eficiência do combustível em regime estacionário	mL/(KJm/s)
M	Massa do veículo	kg

Este modelo [11, 15] fundamenta-se no anterior, sendo que, neste se passa a considerar que o consumo é dependente da aceleração, tornando-o, assim, um modelo mais "robusto". Como é um modelo baseado no anterior, sofre dos mesmos problemas deste, tendo, apesar de tudo, uma maior aproximação à realidade.

2.2.1.5 Modelo 5

$$IFC = \max(\alpha, \beta(P_{out} - P_{eng})) \quad (2.9)$$

$$\beta = \beta_b(1 + eh p \frac{P_{out}}{P_{max}}) \quad (2.10)$$

$$P_{acs} = EALC \frac{RPM}{TRPM} + ECLFC P_{max} \left(\frac{RPM}{TRPM} \right)^{2.5} \quad (2.11)$$

$$P_{eng} = ceng \times beng \times \left(\frac{RPM}{1000} \right)^2 \quad (2.12)$$

Tabela 2.13: Saídas do modelo 5

Saída	Descrição	Unidades
IFC	Consumo de combustível	mL/km
β	Eficiência do combustível	mL/(KJm/s)
P_{eng}	Potência para motor funcionar	kW
P_{out}	Potência total usada pelo motor	kW
P_{acs}	Potência dos acessórios	kW

Tabela 2.14: Entradas do modelo 5

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
RPM	Velocidade do motor	rpm	Rede CAN do veículo	-

Tabela 2.15: Parâmetros do modelo 5

Parâmetro	Descrição	Unidades
P_{max}	Potência máxima	kW
α	Consumo de combustível em regime estacionário	mL/s
β_b	Eficiência base do combustível	mL/(KJm/s)
$EALC$	Constante de carga dos acessórios	kW
ehp	Diminuição proporcional na eficiência de alta potência de saída	-
$TRPM$	Carga máxima à velocidade máxima do motor	-
$ECLFC$	Constante da ventoinha de arrefecimento	-
$ceng$	Parâmetro independente da velocidade de arrasto do motor	-
$beng$	Parâmetro independente da velocidade de arrasto do motor	-

Neste modelo [11, 16] considera-se que o consumo depende do poder de tração do veículo e da potência do motor que é função da respetiva velocidade de rotação. Assim, este modelo torna-se interessante quando existe acesso à velocidade de rotação do motor e quando todos os parâmetros para a sua calibração são conhecidos.

2.2.1.6 Modelo 6

$$F(t) = \begin{cases} \alpha_1 + \beta_1 P(t) & P(t) > 0 \\ \alpha_1 & P(t) \leq 0 \end{cases} \quad (2.13)$$

Tabela 2.16: Saídas do modelo 6

Saída	Descrição	Unidades
$F(t)$	Taxa de consumo de combustível instantâneo	dm^3/s

Tabela 2.17: Entradas do modelo 6

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
$P(t)$	Potência de tração instantânea	kW	Sensor específico	-
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro

Tabela 2.18: Parâmetros do modelo 6

Parâmetro	Descrição	Unidades
α_1	Taxa de consumo em ponto morto	dm^3/s
β_1	Taxa de consumo de combustível do veículo	$dm^3/s/kW$

Este modelo [10, 17], baseado na potência instantânea pedida, é um modelo no qual se podem encontrar as mesmas considerações do modelo 3. É importante referir que os parâmetros α_1 e β_1 variam com o tempo e com as condições do veículo.

2.2.1.7 Modelo 7

$$IFC = f(P_{tr}, P_{accs} + P_{eng} = \max(\alpha, \xi \times P_{tot} \times (1 + dFuel)) \quad (2.14)$$

$$\xi = \xi_b (1 + ehp \frac{P_{tot} - P_{eng}}{P_{max}}) \quad (2.15)$$

$$(2.16)$$

Tabela 2.19: Saídas do modelo 7

Saída	Descrição	Unidades
IFC	Consumo instantâneo de combustível	mL/s
ξ	Eficiência do motor	$mL/kW/s$

Tabela 2.20: Entradas do modelo 7

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
$dFuel$	Excesso de combustível devido a congestionamento	-	Sensor específico	-
P_{tr}	Potência necessária para ultrapassar forças internas do motor	kW	Sensor específico	-
P_{accs}	Potência dos acessórios	kW	Sensor específico	-
P_{eng}	Constante de carga dos acessórios	kW	Sensor específico	-

Tabela 2.21: Parâmetros do modelo 7

Parâmetro	Descrição	Unidades
ehp	Cavalos do motor	-
ξ_b	Eficiência do motor, dependendo do tipo	-
P_{max}	Potência nominal do motor	kW
α	Consumo do combustível em ponto morto	mL/s

Este modelo é denominado HDM-4 e é apresentado em [11, 18]. Trata-se de um modelo que depende de variáveis de entrada bastante complexas de obter, pelo que se torna bastante difícil a sua implementação.

Concluída a análise efetuada acerca dos diferentes modelos físicos constantes na literatura, na próxima secção realiza-se uma análise semelhante para os modelos empíricos.

2.2.2 Modelos Empíricos

Esta secção é dedicada à apresentação dos modelos empíricos ou de regressão. Estes distinguem-se dos anteriores pois para o seu uso é necessário realizar algumas experiências prévias para obter determinados parâmetros. À semelhança da secção anterior, serão apresentadas referências bibliográficas onde se poderão encontrar informações mais detalhadas.

2.2.2.1 Modelo 8

$$F(t) = \alpha_2 + \beta_a P_a(t) + \beta_b P_c(t) \quad (2.17)$$

$$(2.18)$$

Tabela 2.22: Saídas do modelo 8

Saída	Descrição	Unidades
$F(t)$	Taxa de consumo de combustível instantâneo	dm^3/s

Tabela 2.23: Entradas do modelo 8

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
$P_c(t)$	Potência total exercida pelo motor a velocidade constante	kW	Sensor específico	-
$P_a(t)$	Potência total exercida pelo motor e a sua inércia	kW	Sensor específico	-

Tabela 2.24: Parâmetros do modelo 8

Parâmetro	Descrição	Unidades
α_2	Taxa de consumo em marcha lenta	dm^3/s
β_a	Parâmetros específico da potência do veículo	$dm^3/s/kW$
β_b	Parâmetros específico da potência do veículo	$dm^3/s/kW$

Este modelo, baseado no modelo 6 do Australian Road Research Board[10], apresenta entradas bastantes difíceis de obter. Inclui, ainda, alguns parâmetros que, apesar de estar descrito o que significam, necessitam de ser conseguidos através de regressão.

2.2.2.2 Modelo 9

$$F(t) = \begin{cases} \frac{N}{60000n_r} w_e(t) f_p(t) (a_\delta w_e^2 + b_\delta w_e + c_\delta) & G \neq 0 \\ F_i d l e & G = 0 \end{cases} \quad (2.19)$$

Tabela 2.25: Saídas do modelo 9

Saída	Descrição	Unidades
$F(t)$	Taxa de consumo de combustível instantâneo	dm^3/s

Tabela 2.26: Entradas do modelo 9

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
f_p	Controlo do pedal [0,1]	kW	Rede CAN do veículo	-
G	Sinal da mudança	-	Rede CAN do veículo	-
n_r	Nº de rotações cambota por ciclo	rpm	Sensor específico	-
w_e	Velocidade do motor	rpm	Rede CAN do veículo	-

Tabela 2.27: Parâmetros do modelo 9

Parâmetro	Descrição	Unidades
N	Número de cilindros do motor	-
F_{idle}	O consumo permanente do motor	dm^3/s
β_b	Parâmetros específico da potência do veículo	$dm^3/s/kW$
$a_\delta, b_\delta, c_\delta$	Constantes obtidas experimentalmente	-

Este modelo pretende simular o sistema, tendo por base a posição do pedal do acelerador e do pedal do travão, bem como a mudança em que o carro se encontra e a velocidade do motor[10, 19]. Para além de exigir acesso à rede CAN do veículo, exige, ainda, a introdução de sensores específicos para obter algumas entradas. Requer, também, várias experiências para se obterem determinados parâmetros.

2.2.2.3 Modelos 10-11

$$F_1(t) = \alpha_0 + \alpha_1 P(t) + \alpha_2 P(t)^2 \quad (2.20)$$

$$F_2(t) = \beta_0 + \beta_1 P(t) + \beta_2 P(t)^2 \quad (2.21)$$

$$P(t) = \frac{R(t) + 1.04ma(t)}{3600\eta_d} v(t) \quad (2.22)$$

$$R(t) = \frac{\rho}{25.92} c_d C_H(t) A_f v^2(t) + 9.8066mC_r[c_1 v(t) + c_2] + 98066mG(t) \quad (2.23)$$

Tabela 2.28: Saídas do modelo 10 e 11

Saída	Descrição	Unidades
$F_1(t), F_2(t)$	Taxa de consumo de combustível instantâneo	dm^3/s

Tabela 2.29: Entradas do modelo 10 e 11

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
$P(t)$	Potência pedida	kW	Sensor específico	-
$v(t)$	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro
$a(t)$	Aceleração do veículo	m/s^2	Acelerómetro	GPS
$R(t)$	Forças de resistência exercidas no veículo	N	Sensor específico	-

Tabela 2.30: Parâmetros do modelo 10 e 11

Parâmetro	Descrição	Unidades
m	Massa do veículo	kg
C_d	Coeficiente de arrastamento do ar	-
C_H	Coeficiente de correção de altitude	-
A_f	Área frontal do veículo	m^2
G	Força gravítica	m/s^2
C_r, c_1, c_2	Parâmetros dependentes da resistência ao rolamento	-
$\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2$	Constantes específicas do veículo	-

Estes modelos são denominados "Comprehensive Power-based Fuel Consumption Models 1"(CPMF-1) e "Comprehensive Power-based Fuel Consumption Models 2"(CPMF-2) [10, 19], respetivamente, sendo a sua diferença a calibração da variável β_0 que no CPMF-2 é dependente da velocidade do motor, enquanto que todas as outras variáveis são obtidas através de ciclos de condução normalizados. De notar que, algumas das entradas necessitam de sensores específicos para a sua medição.

2.2.2.4 Modelo 12

$$FC = \frac{1}{FE} \quad (2.24)$$

$$FE = \beta_0 + \beta_1 P(t) + \beta_2 P(t)^2 \quad (2.25)$$

$$P(t) = a \left(\frac{T}{2} + b \right)^2 \quad (2.26)$$

Tabela 2.31: Saídas do modelo 12

Saída	Descrição	Unidades
FC	Consumo de combustível	L/km
FE	Economia de combustível	km/L

Tabela 2.32: Entradas do modelo 12

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
T	Binário do motor	Nm	Sensor específico	-

Tabela 2.33: Parâmetros do modelo 12

Parâmetro	Descrição	Unidades
a, b, c	Coeficientes de regressão, dependentes da mudança	kg

Este modelo é bastante simples, tratando-se de uma simples regressão linear desenvolvida por um estudo realizado pela USDOT[11, 20]. Apresenta os mesmos problemas que o anterior relativamente às entradas, bem como relativamente aos parâmetros necessários para o seu funcionamento.

2.2.2.5 Modelo 13

$$\ln(MOE) = \begin{cases} \sum_{i=0}^3 \sum_{j=0}^3 (L_{i,j}^e \times s^i \times a^j) & a \geq 0 \\ \sum_{i=0}^3 \sum_{j=0}^3 (M_{i,j}^e \times s^i \times a^j) & a < 0 \end{cases} \quad (2.27)$$

Tabela 2.34: Saídas do modelo 13

Saída	Descrição	Unidades
<i>MOE</i>	Consumo de combustível instantâneo ou taxa de emissão	<i>L/s</i> ou <i>mg/s</i>

Tabela 2.35: Entradas do modelo 13

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
<i>s</i>	Velocidade do veículo	km/h	Rede CAN do veículo	GPS / Acelerómetro
<i>a</i>	Aceleração do veículo	km/h/s	Acelerómetro	GPS

Tabela 2.36: Parâmetros do modelo 13

Parâmetro	Descrição	Unidades
<i>e</i>	Indicador do tipo de consumo / tipo de emissão	-
<i>Lei, j / Mei, j</i>	Coefficientes de regressão, dependentes da mudança	-

Este modelo, presente em [21, 22], é denominado VT-MICRO e permite, calibrando com os índices indicados, obter uma estimativa para consumo de combustível, bem como emissões de CO_2 . Saliente-se que, devido às entradas deste modelo serem de obtenção simples, é de considerar a sua utilização.

2.3 Sistema de Aquisição de Dados Vocacionados para Automóveis

Tendo em consideração o problema do armazenamento de dados no dispositivo móvel para uma análise posterior, sentiu-se a necessidade de estudar alguns sistemas existentes que realizassem essa mesma função. Assim, nesta secção, serão expostos alguns sistemas de aquisição de dados, capazes de armazenarem variáveis que podem ser interessantes no âmbito desta dissertação.

- **DL1**

Uma análise deste sistema de aquisição pode ser encontrada em [23]. Trata-se de um sistema de aquisição para automóveis com o intuito de guardar velocidades e acelerações a que este está sujeito. Uma das suas características mais relevante é a de possuir um recetor de GPS, permitindo, por isso, intercalar as medidas obtidas pelos sensores internos com as medidas obtidas externamente, melhorando, assim, a qualidade dos dados. Estes são guardados num cartão de memória *compact flash*.

- **Digitek - Cobra**

Este sistema de aquisição, analisado em [23], não possui qualquer tipo de sensor interno. Contudo, é capaz de amostrar dados a uma frequência de 1kHz. Possui 64 Mb de memória interna para guardar dados.

- **Digitek - Shark**

O sistema de aquisição, presente em [23], é em tudo semelhante ao anterior, possuindo, porém, capacidade para amostrar dados a 4kHz e, ainda, 256 Mb de memória interna, podendo ser expandido com um cartão de memória de 1Gb.

- **SEAL platform**

Trata-se de um sistema de aquisição de baixo custo, desenvolvido em [24] com o objetivo de monitorizar o estado de um automóvel. Este, em que os dados são armazenados num cartão de memória *flash*, inclui um recetor GPS para que exista. Os sensores básicos deste dispositivo incluem, entre outros, acelerómetro de dois eixos e medidor de campo magnético em dois eixos, bem como o GPS já referenciado.

- **Sistema de Aquisição de Dados de Baixo Custo**

Foram encontrados outros sistema de aquisição em [25, 26], que, de forma semelhante aos sistemas anteriores, pretendem, com um baixo custo, monitorizar o estado do automóvel através de sensores simples, como acelerómetros, giroscópios e GPS.

Capítulo 3

Análise dos Modelos Matemáticos

Nesta parte serão estudados os modelos que parecem ser os mais promissores. Assim, esta debruça-se sobre dois tipos de modelos, um físico e um empírico, com o propósito de se avaliar qual o melhor método para estimar o consumo de combustível. Igualmente, apresenta-se a ferramenta Advisor(*Advanced Vehicle Simulator*)[27], a qual permitiu validar os referidos modelos, simulando o comportamento de um veículo real.

3.1 Advisor: Ferramenta para Validação dos Modelos

A opção por esta ferramenta fundamentou-se numa pesquisa acerca dos diferentes tipos de mecanismos existentes para simulação de veículos automóveis, a qual conduziu ao estudo da ferramenta Advisor[27], mais especificamente do modelo Simulink denominado BD_CONV. Este pretende simular o comportamento de um automóvel convencional. O resultado desta análise é apresentado de uma forma sucinta nesta secção, tendo por objetivo exemplificar de que modo se pode simular o consumo de combustível para vários ciclos de condução, podendo estes ser ou os já encontrados na plataforma ou os introduzidos pelo utilizador.

3.1.1 Principais Blocos do Modelo BD_CONV

Nesta secção será apresentado o modelo BD_CONV, explicitando, detalhadamente, cada um dos seus blocos.

Na figura 3.1 é possível ter uma visão global do diagrama Simulink, enumerado-se e analisando-se, de seguida, os seus sub-blocos

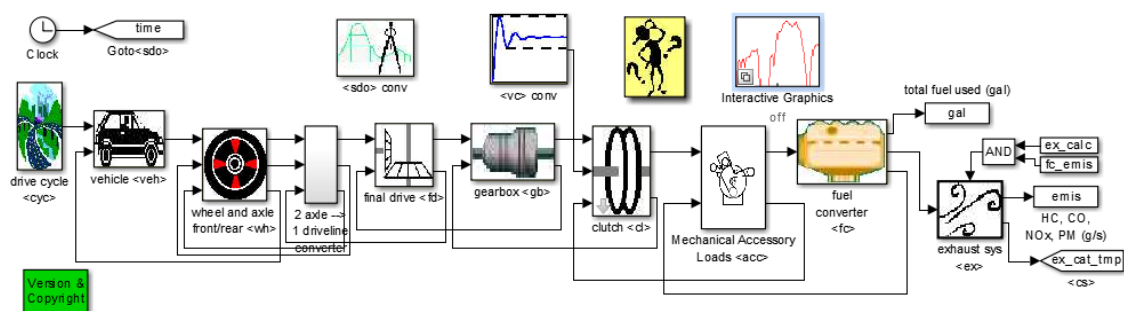


Figura 3.1: Visão Geral do Modelo BD_CONV[27]

- vehicle <veh>

O módulo, apresentado na figura 3.2, converte a velocidade pedida (no início e no fim do *step*), a força de tração e a velocidade que podem ser atingidas no momento final, em força de tração e velocidade precisas nos pneus. Este módulo inclui os seguintes blocos:

- **rolling resistance force.** Com base na velocidade do veículo, calcula as forças resistivas devido ao rolamento. Depende dos coeficientes de rolamento e do ângulo da estrada;
- **force req'd to ascend.** Com base na velocidade atual, se maior que 0, calcula a força necessária para subir, atendendo à inclinação da estrada, à gravidade e à massa do veículo;
- **force req'd to overcome aerodynamic drag.** Com base no quadrado da velocidade e nos coeficientes relacionados com a densidade do ar, calcula a força para ultrapassar forças aerodinâmicas;
- **force req'd to accelerate.** Com base nas velocidades, anterior e atual, bem como no tempo decorrido entre amostras, calcula a aceleração e a força necessárias para mover o carro, considerando a sua massa;
- **vehicle speed.** Este bloco contempla a velocidade no instante anterior e a força de tração existente, calculando a velocidade no final do *step* de amostragem. Tal é efetuado resolvendo uma equação de segundo grau do tipo $ax^2 + bx + c$, onde:

- * *a*: Dado de acordo com os coeficientes de arrasto de ar.
- * *b*: Este bloco depende da inclinação da estrada, do tempo de amostragem e do carro estar ou não em movimento. É, ainda, determinado pela massa do veículo.
- * *c*: Este subsistema depende dos mesmos fatores que *b* acrescentando-lhe outros fatores, tais como a força de tração e a velocidade anterior. Nele considera-se, ainda, a gravidade a que o veículo está sujeito, entre outros fatores de menor relevância. O resultado final é uma força de tração.

Através da resolução da equação e tendo por base a velocidade linear que se pode atingir no máximo, obtém-se a velocidade no final do tempo de amostragem.

Este sistema, por fim, coloca na saída a força pedida pela roda e pelo pneu.

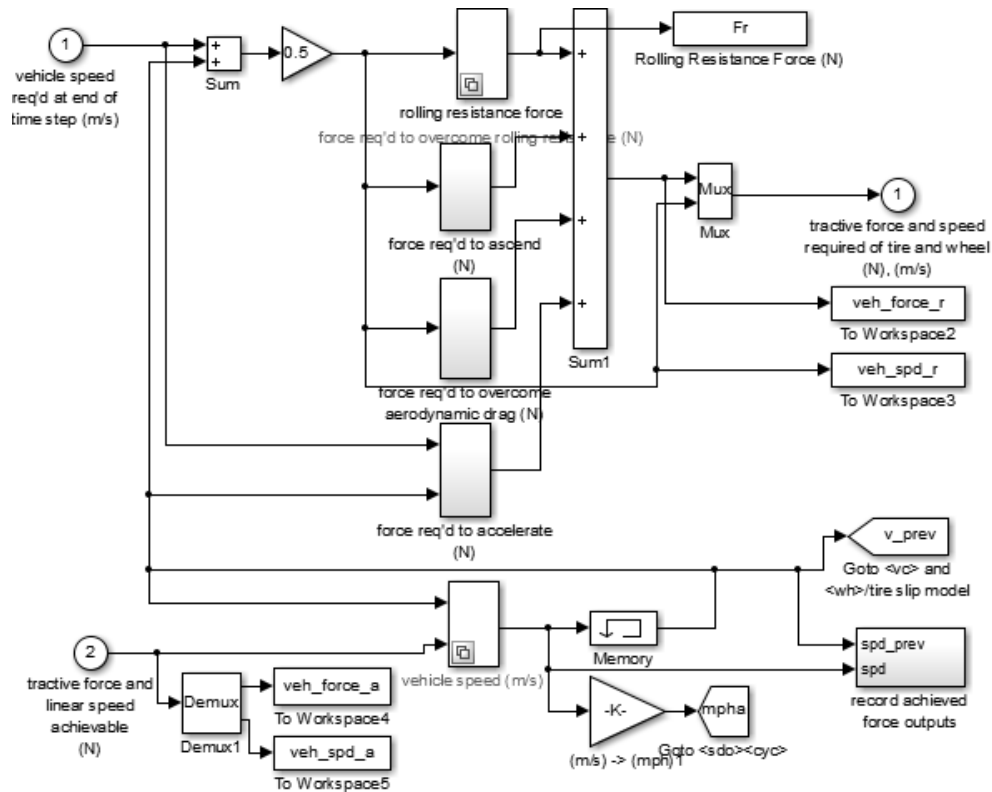


Figura 3.2: Bloco vhe[27]

- wheel and axle front/rear <wh>

Este bloco tem como entrada (*input*) a saída do bloco anterior, ou seja, a força de tração, a velocidade requerida pelos pneus e a interface com a estrada, ao que se associam os binários e a velocidade disponíveis do sistema de transmissão para ambos os eixos. Como saídas (*output*), tem o binário e a velocidade precisa do sistema de transmissão para ambos os eixos, assim como a força e a velocidade linear atingível. Inclui diferentes (sub)blocos que se passam a enunciar:

- **traction controller interface.** Com base na força de tração e na velocidade requerida pela interface com a estrada, disponibiliza essas mesmas variáveis á saída, mas já limitadas, as velocidades angulares dos eixos frontal e traseiro, assim como o rácio entre o uso dos eixos;

- **tire slip model.** Este bloco baseia-se nas saídas do bloco anterior e coloca na saída o deslizamento dos pneus, considerando o pneu e a estrada. A soma dos 3 blocos seguintes disponibiliza o binário pedido do sistema de tração para o eixo frontal;
- **effect of inertia.** Trata-se de um bloco que calcula o binário necessário para ultrapassar a inércia do conjunto rodas e veículo, tendo por base a velocidade do eixo;
- **loss.** De acordo com a posição do eixo e a sua velocidade, este bloco calcula o binário perdido ;
- **front force.** Bloco que calcula o binário requerido pelo sistema de transmissão;

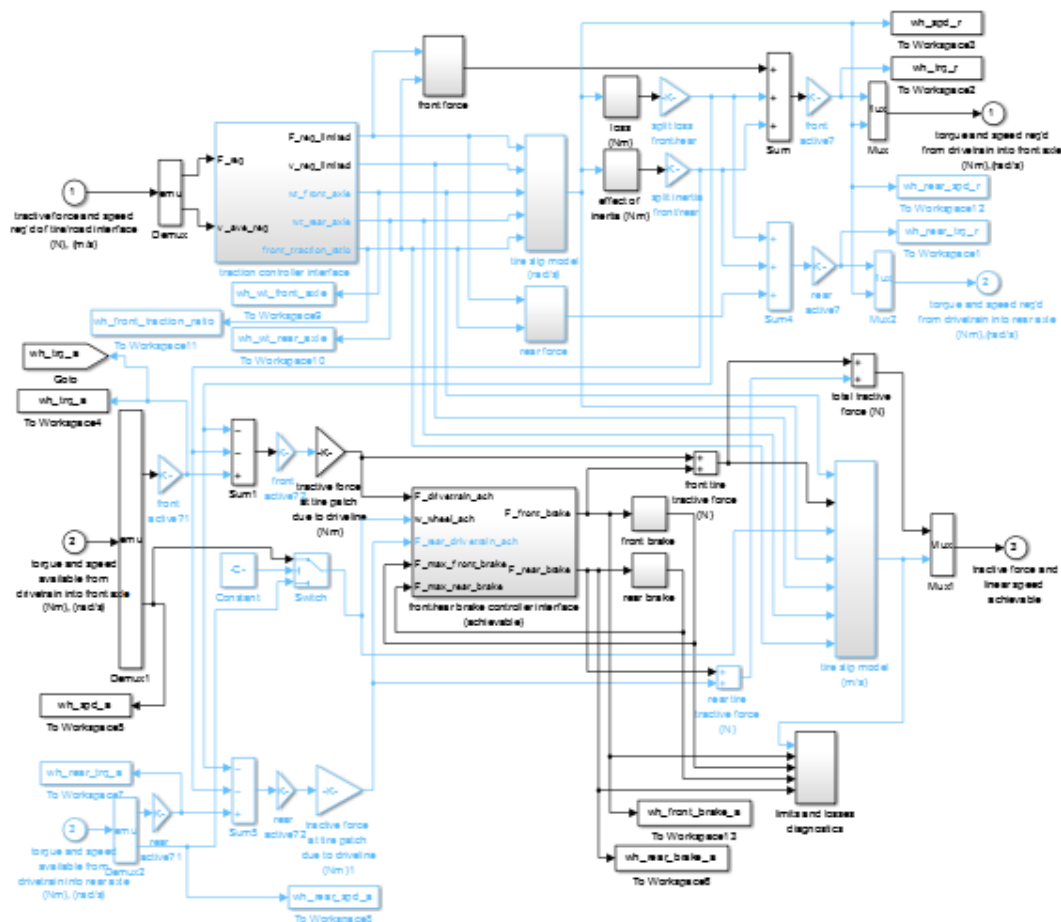


Figura 3.3: Bloco wh[27]

- axle → 1 driveline converter

A função deste bloco é a de converter as necessidades de binário (anterior e posterior) e a velocidade em um único requisito. Converte, ainda, as necessidades de tração e a velocidade do final drive em necessidades para os dois eixos (anterior e posterior). A primeira saída entra no final drive, enquanto que as outras duas saídas se destinam ao bloco anterior (wheel

and axl front/rear). Estas operações são executadas tendo por base operações simples, tais como *switchs*, somas e multiplicações.

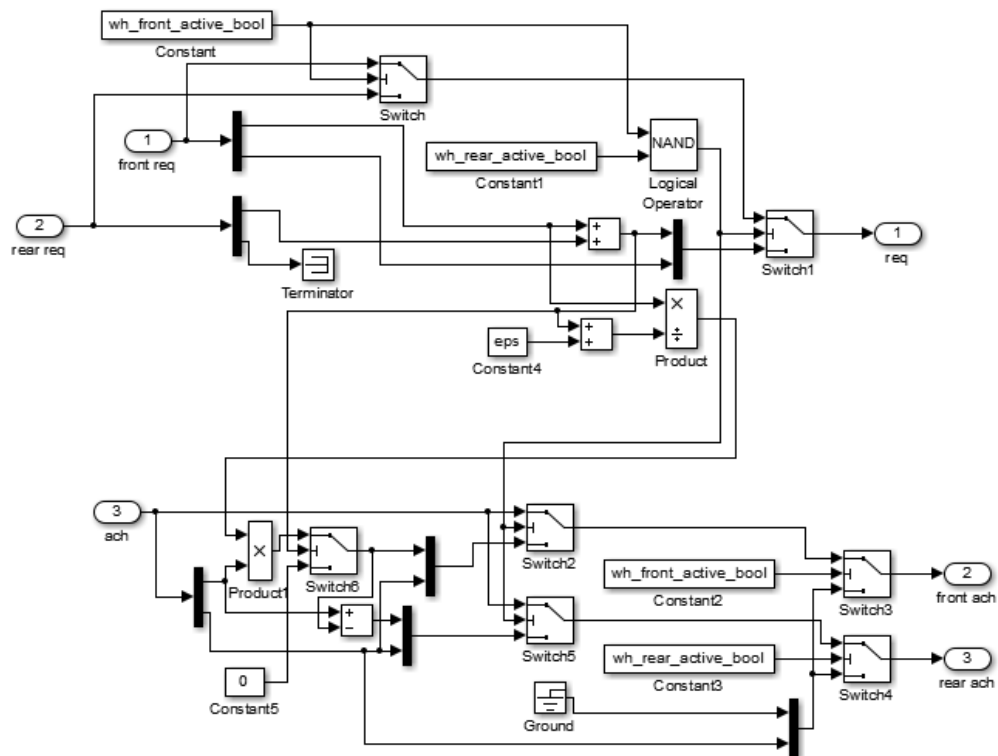


Figura 3.4: Bloco axle[27]

- final drive <fd>

Este bloco tem por função converter as necessidades de binário e a velocidade do lado de saída do *drive* para o lado da entrada. Converte, também, o binário e a velocidade disponíveis na entrada do final drive para a saída. Usando, à semelhança do bloco anterior, operações simples como, por exemplo, *switchs*, somas e multiplicações.

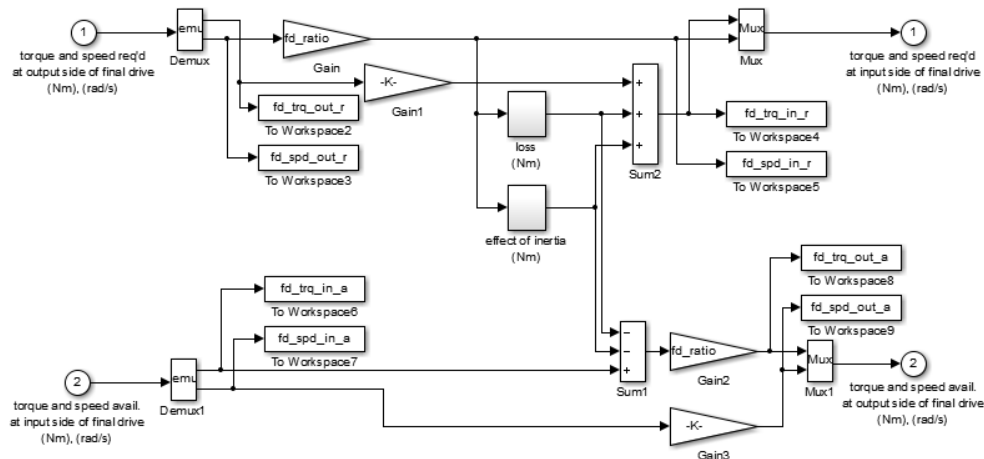


Figura 3.5: Bloco fd[27]

- gearbox <gb>

Este bloco, que simula a caixa de velocidades, tem por função converter as necessidades de binário e a velocidade do lado de saída da caixa de velocidades para o lado da entrada. Similarmemente, converte o binário e a velocidade disponíveis na entrada da caixa de velocidades para a saída.

- **gearbox controller interface.**
- **Pout/Pin neglecting inertia.** Este sub-bloco calcula o rácio entre o binário de saída e o binário de entrada, com mecanismos para evitar divisões por zero.

A segunda conversão é executada com recurso ao bloco gearbox controller interface e a blocos mais simples, que apenas executam operações de multiplicação, de divisão e de soma.

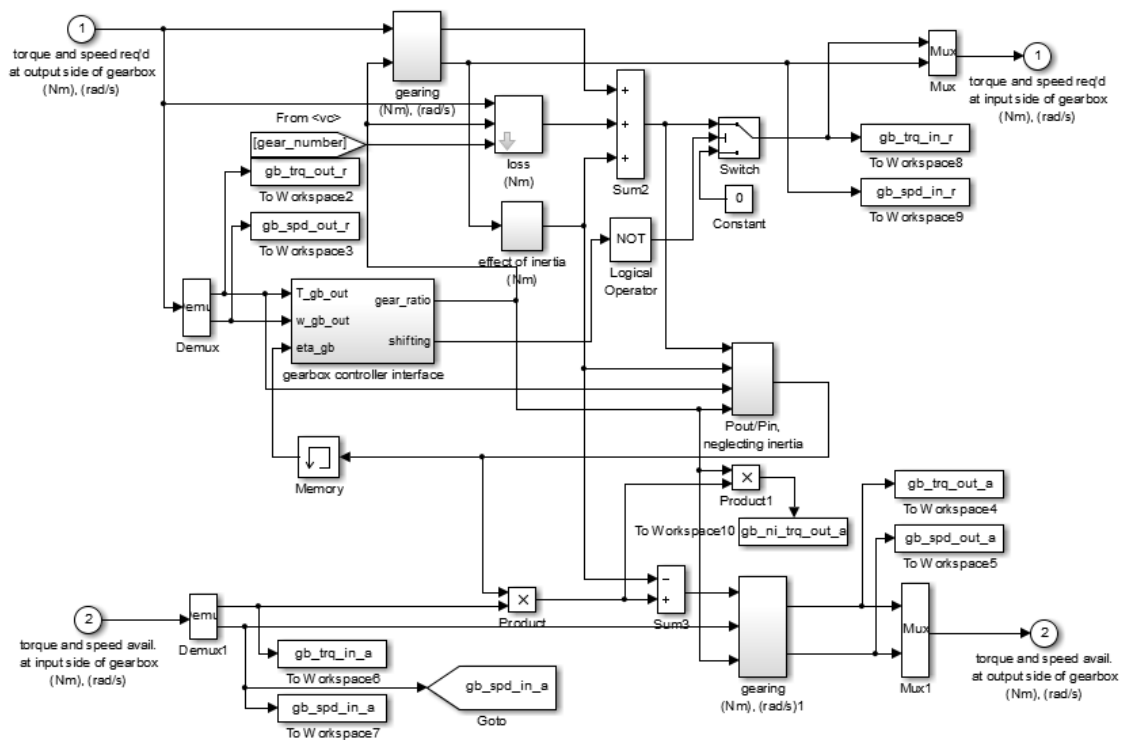


Figura 3.6: Bloco gb[27]

- clutch <cl>

Este bloco simula a embraiagem. Como tal, tem como entrada o binário e a velocidade requeridos na sua saída, a posição da embraiagem e o binário disponível na saída. Apresenta, como saída, o binário e a velocidade requeridas à entrada e os disponíveis na sua saída. Estas operações são executadas com recurso a blocos lógicos.

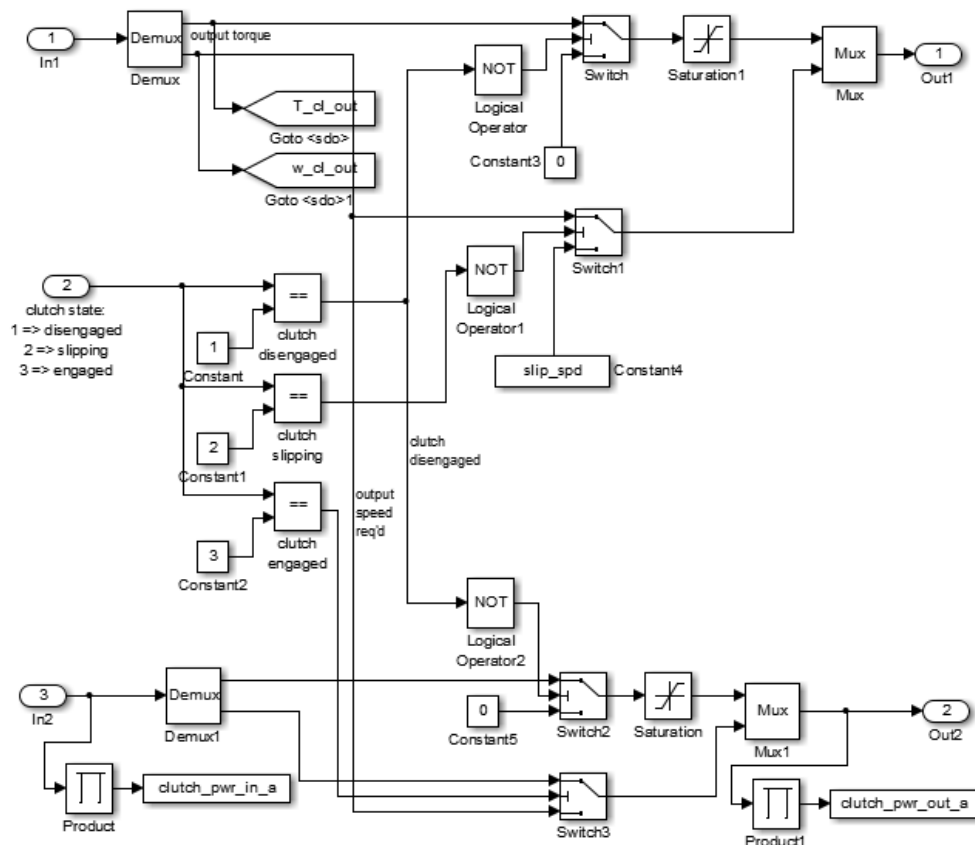


Figura 3.7: Bloco cl[27]

- Mechanical Accessory Loads <acc>

Será tratado como uma caixa preta que converte o binário e a velocidade de saída do eixo em entrada, considerando as cargas. Converte, também, o binário e as velocidades atingíveis do eixo para o binário, bem como a velocidade para o sistema de transmissão. Estas considerações são executadas considerando as cargas acessórias ligadas/desligadas.

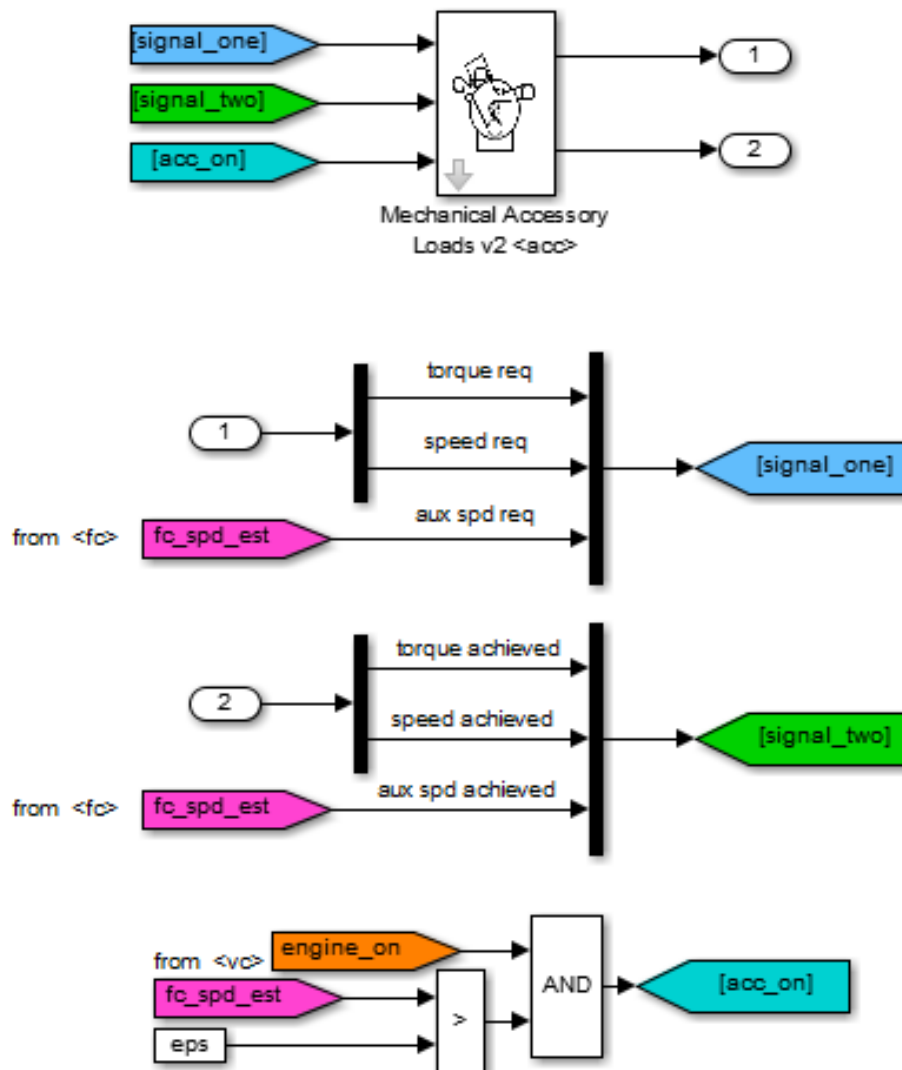


Figura 3.8: Bloco acc[27]

- fuel converter <fc>

Este bloco tem como entrada o binário e a velocidade requeridos no motor, colocando na saída o uso de combustível, as emissões de gases, o binário e a velocidade disponíveis para o sistema de transmissão. Nele existem diversos sub-blocos, a saber:

- **engine torque.** Com base no binário pedido, na velocidade do motor estimada e no controlo de velocidade, este bloco coloca na saída o binário pedido pelo motor. Consideram-se, aqui, os efeitos da inércia (simulada num bloco), e, após isso, através de blocos simples de limitação, somas e *switchs*, coloca na saída o binário para o eixo;

- **engine speed estimator.** Estima a velocidade com base em parâmetros da velocidade requerida pelo ciclo de viagem, pelo estado da embraiagem e com o comando da velocidade;
- **fuel use and EO emis (fuel use and EO emis Configurable Subsystem).** As saídas dos dois blocos anteriores entram neste bloco onde é calculado o consumo de combustível. Este cálculo é efetuado com base em dois blocos, sendo o primeiro, HOT engine maps, aquele em que, através de uma *look up table* e com base no binário e na velocidade, calcula o HOT fuel use. Seguidamente, entra num bloco denominado fc temp correct, onde, com base na temperatura correta, se corrige o consumo de combustível multiplicando-o por um fator dependente da temperatura.

Com base neste consumo instantâneo, pode-se proceder à integração do mesmo para que se conheça o consumo total.

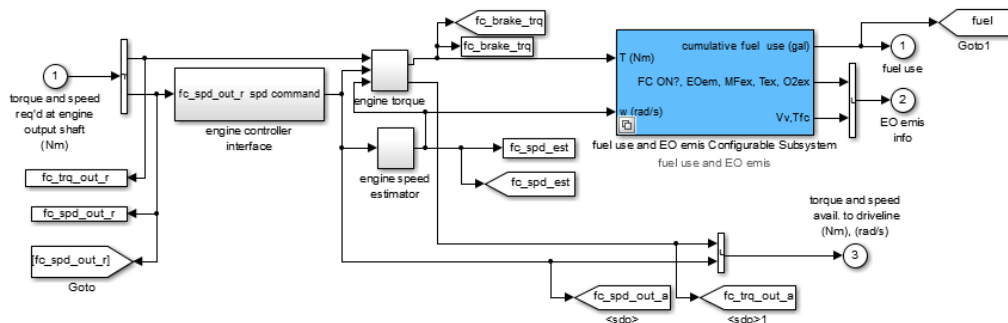


Figura 3.9: Bloco fc[27]

- exhaust sys <ex>

Este bloco tem por base a saída do bloco anterior e calcula as emissões de HC, CO, NOx e Pm do automóvel.

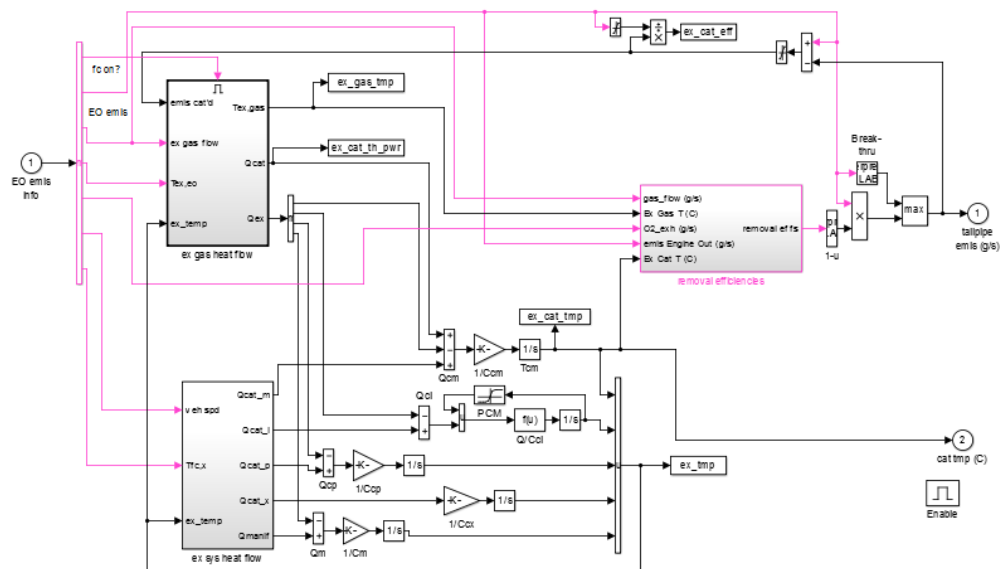


Figura 3.10: Bloco ex[27]

- **<vc> conv**

Neste bloco é onde se encontra simulado o controlo do automóvel por parte do condutor, ou seja, estão aqui definidos a forma como é pressionada a embraiagem, o modo como se controla o motor, a tração e a caixa de velocidade e como se processam as travagens.

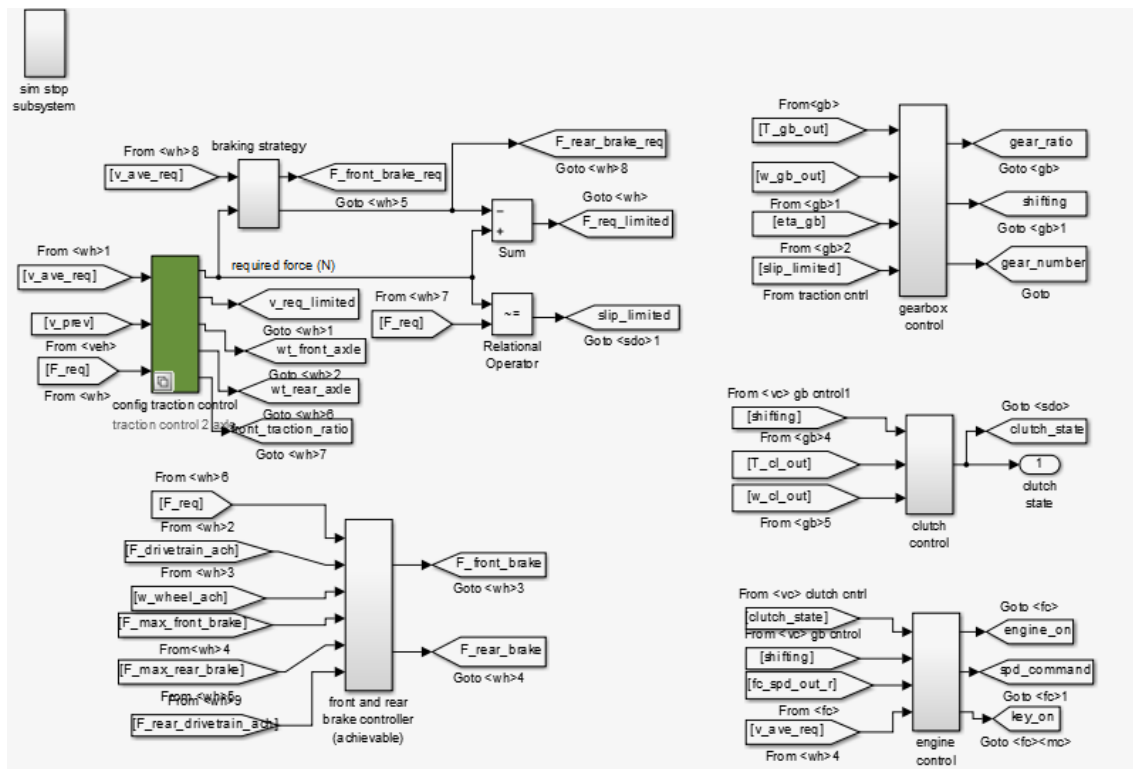


Figura 3.11: Bloco vc[27]

- drive cycle <cyc>

Neste bloco insere-se o ciclo de condução onde estão incluídas as velocidades em função do tempo, sendo que são, ainda, tomadas em linha de conta a carga do veículo e a altura a que se encontra de acordo com a distância percorrida.

- <sdo> conv

Este bloco pretende servir de interface para a plataforma Advisor.

3.1.2 Interface

Esta ferramenta é executada recorrendo a uma interface gráfica em MATLAB na qual é possível caracterizar um veículo e todos os seus parâmetros, bem como os ciclos de condução a que este estará sujeito. Por fim, apresentam-se os resultados da simulação. Em cada uma das fases intermédias da interface é possível guardar as características pretendidas, para que, posteriormente, possam ser carregadas na ferramenta. De igual modo, é possível introduzir, no Advisor, dados obtidos pelo utilizador, necessitando-se apenas de configurar os mesmos de forma idêntica aos disponibilizados pela ferramenta. Como tal, é possível, por exemplo, introduzir na ferramenta novos ciclos de condução.

De seguida são apresentados figuras com as três etapas do Advisor, acompanhadas por uma breve descrição dos parâmetros modificáveis.

- Definição de veículo

Nesta interface é possível definir o veículo, bem como todos os seus parâmetros. É, também, escolhido qual o modelo para simulação.

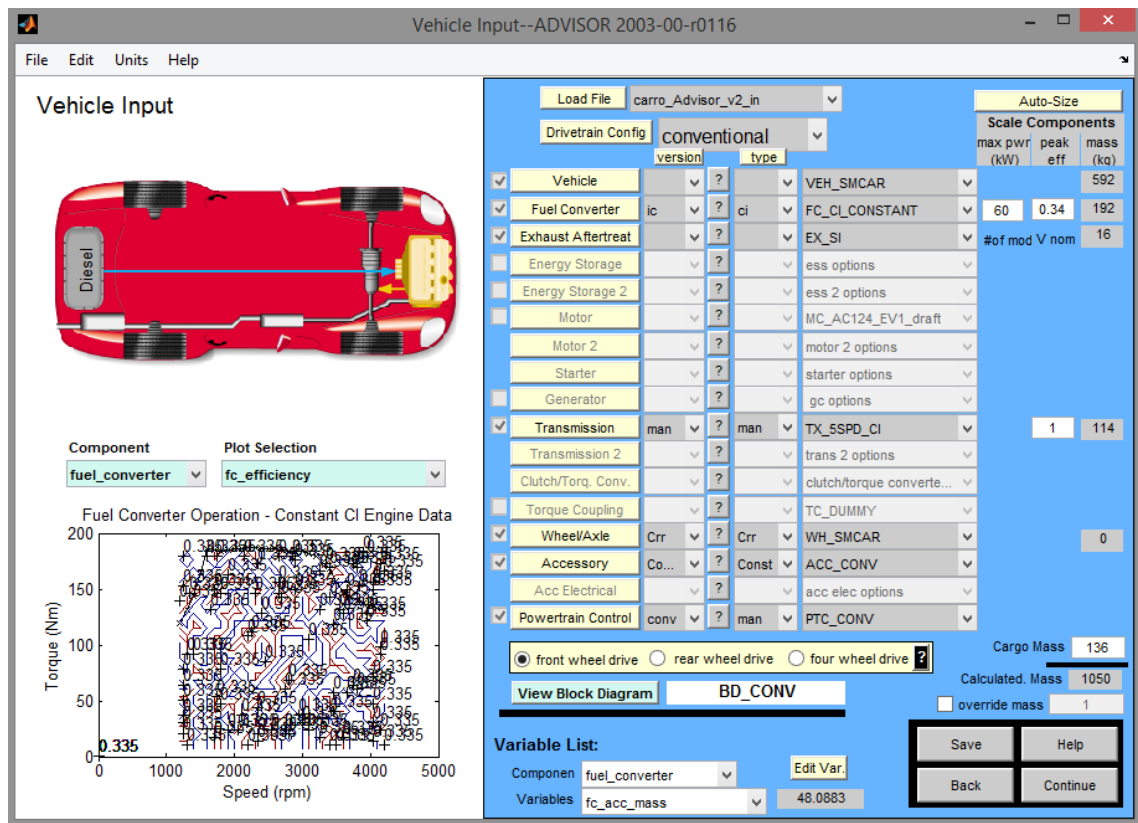


Figura 3.12: Interface Advisor - Definição de veículo[27]

- Definição de ciclo de condução

Neste menu é possível escolher o ciclo de condução a ser percorrido e, da mesma maneira, definir variações em determinados parâmetros, tais como o peso, ao longo do percurso.

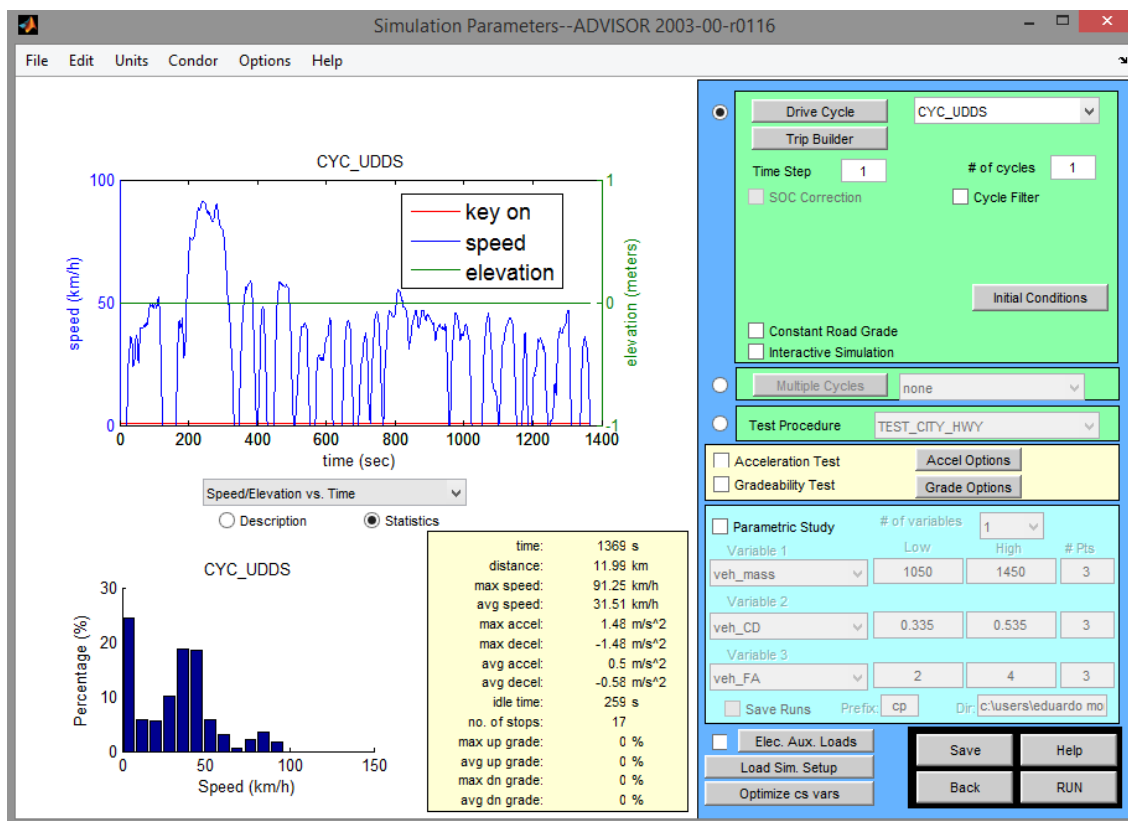


Figura 3.13: Interface Advisor - Definição de ciclo de condução[27]

- Apresentação de resultados

Nesta interface são apresentados os vários resultados, existindo a possibilidade de modificar e de escolher as variáveis a visualizar.

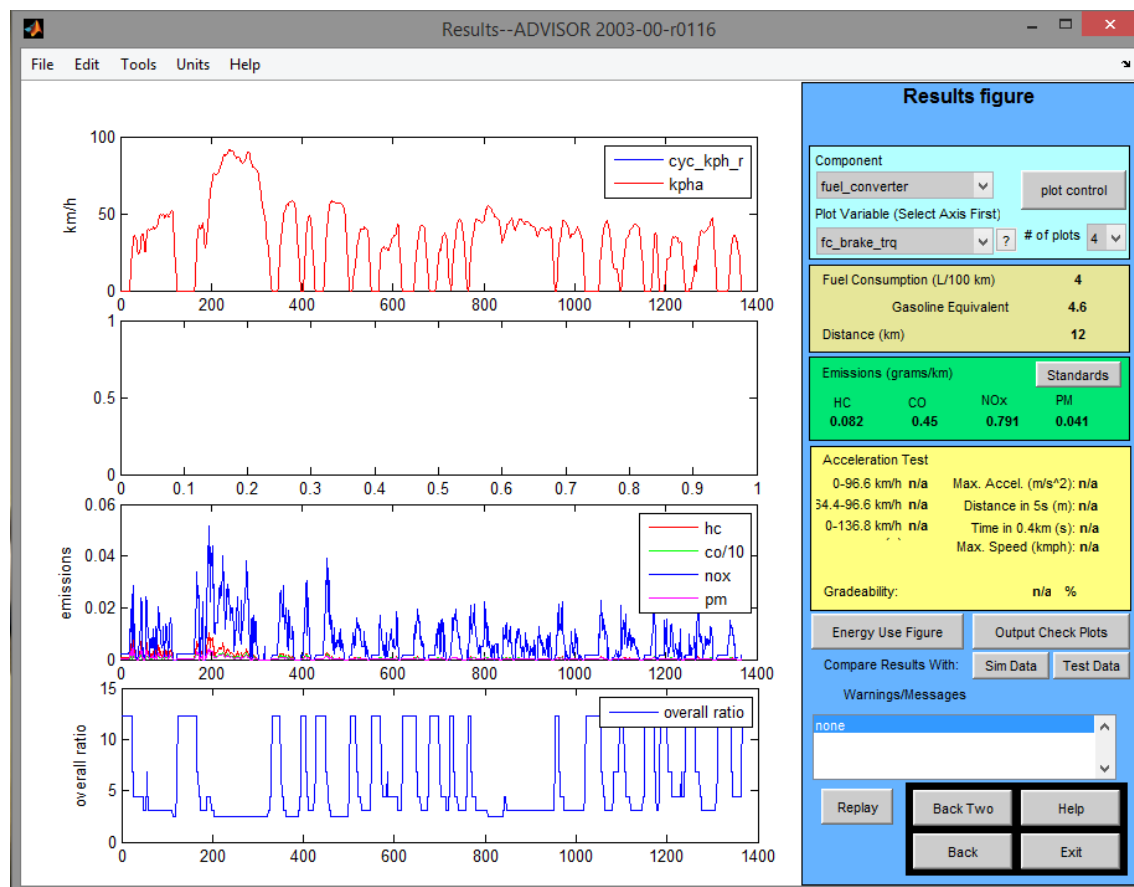


Figura 3.14: Interface Advisor - Apresentação de resultados[27]

3.2 Análise e Calibração de modelos

Nesta secção serão explicados os modelos escolhidos para a estimação do consumo de combustível e, de seguida, será tratada a calibração preliminar dos mesmos, recorrendo à sua simulação em Simulink, comparando os resultados com os obtidos através do Advisor.

3.2.1 Seleção de Modelos para Estimativa

De modo a seleccionar os modelos mais adequados para estimar o consumo de combustível, concretizou-se uma análise comparativa dos mesmos, na qual se consideraram os seguintes parâmetros: facilidade para obter entradas; facilidade para calibrar; e elementos considerados.

O método de comparação dos vários modelos é apresentado na seguinte tabela. Nesta usa-se uma escala de -, -, +, ++, em que - significa muito difícil ou poucos e ++ significa de fácil acesso ou muitos elementos considerados. Refira-se, ainda, que se efetuou uma análise para os modelos físicos e outra para os modelos empíricos.

Tabela 3.1: Comparação entre modelos físicos

Modelo	Facilidade para obter entradas	Facilidade para Calibrar	Elementos considerados
1	-	-	++
2	++	-	++
3	-	+	-
4	-	+	-
5	+	-	+
6	-	+	-
7	-	-	+

Tabela 3.2: Comparação entre modelos empíricos

Modelo	Facilidade para obter entradas	Facilidade para Calibrar	Elementos considerados
8	-	+	-
9	-	-	+
10/11	-	-	++
12	-	+	-
13	++	-	+

Face às tabelas apresentadas, concluiu-se que para o trabalho visado, isto é, o desenvolvimento de um algoritmo para estimar consumo de combustível em tempo real, o modelo físico mais adequado a utilizar seria o número 2, enquanto que, no caso empírico seria o modelo 13.

3.2.2 Calibração dos Modelos

Neste ponto serão abordadas a parametrização e a calibração dos modelos 2 e 13. Esta análise será concretizada tendo em consideração parâmetros, consumos e ciclos de condução extraídos da plataforma Advisor. Pretende-se, assim, comparar de que forma estes dois modelos se comportam, quando comparados com o modelo mais complexo usado pela plataforma.

Inicia-se pela realização de uma análise do modelo 2, verificando as equações deste modelo, já apresentadas em 2.5, as variáveis deste modelo podem ser divididas em três tipos: entradas, saídas e parâmetros, do veículo e do ambiente. Na primeira etapa da análise serão avaliados os parâmetros do modelo. Pelo que, serão usados parâmetros extraídos da plataforma Advisor, para que, na etapa de subsequente de simulação seja possível comparar corretamente os resultados.

3.2.2.1 Modelo 2 - Obtenção dos Parâmetros Físicos do Advisor

Considerando a lista, previamente apresentada, dos parâmetros do modelo, é possível obter diretamente do Advisor os seguintes parâmetros:

Tabela 3.3: Parâmetros obtidos diretamente do Advisor

Parâmetro	Valor	Unidades
g	9.81	m/s^2
A	2	m^2
ρ	1.225	kg/m^3
C_d	0.009	—
C_r	0.335	—

É, ainda, possível calcular H_g recorrendo às variáveis do Advisor. Assim, com recurso às variáveis valor calorífico do combustível (J/g) e à sua densidade (g/L), é possível obter H_g nas unidades pretendidas.

Tabela 3.4: Parâmetros obtidos indiretamente do Advisor

Parâmetro	Valor	Unidades
H_g	36196	J/cm^3

3.2.2.2 Modelo 2 - Obtenção de Parâmetros Recorrendo a Calibração

Para uma completa calibração do modelo, é necessário obter um valor para parâmetros que não estão diretamente presentes no Advisor, sendo eles o consumo em ponto morto (ft_{idle}), a eficiência de travagem e de transmissão (η_t, ϵ). Todavia, o primeiro não existe como variável e os restantes são variáveis dependentes de muitos outros parâmetros, pelo que são usados dois processos diferentes. É, ainda, necessário conhecer o valor das massas (m, m_j) presentes nas equações do modelo. Estas foram consideradas aproximações do peso total presente no Advisor, uma vez não ser possível obter, direta ou indiretamente, o seu valor no Advisor.

Tabela 3.5: Massas obtidas através de aproximação

Parâmetro	Valor	Unidades
m	1000	kg
m_j	50	kg

Para ft_{idle} , usou-se um ciclo de condução com velocidade nula e verificou-se qual o consumo respetivo, obtendo-se assim ft_{idle} .

Para obtenção dos restantes parâmetros, considerou-se que, devido a estes se apresentarem na estimativa como um produto, uma das eficiências toma o valor 1, variando apenas a outra. Por este motivo, realizaram-se várias simulações do modelo em Advisor para ciclos de condução com velocidades constantes diferentes de zero. Posteriormente, simulou-se múltiplas vezes o modelo 2, com os mesmos ciclos de condução, variando, apenas, o valor da eficiência, a fim de se obterem consumos iguais no mesmo ciclo de condução. Os valores obtidos são apresentados na tabela seguinte:

Tabela 3.6: Parâmetros obtidos através de calibração

Parâmetro	Valor	Unidades
f_{idle}	0.0950	cm^3/s
ϵ	1	—
η_t	0.24	—

3.2.2.3 Modelo 13 - Obtenção de Parâmetros Recorrendo a Calibração

Devido ao carácter empírico do modelo 13, tornou-se mais difícil a obtenção de resultados favoráveis. Apesar de se ter conseguido obter resultados com curvas semelhantes às presentes no Advisor, tornou-se demasiado trabalhoso calibrar o modelo. Daí, tendo em consideração o facto de que, até ao momento, já se tinham obtido resultados favoráveis na calibração do modelo 2, decidiu-se usar apenas este último, visto ser mais simples e os seus resultados serem suficientemente bons.

As dificuldades relatadas na calibração do modelo 13 ilustram-se na seguinte figura. Nela é possível visualizar que os resultados da simulação do modelo 2, apesar de não estarem completamente de acordo com os resultados do Advisor, seguem o mesmo tipo de gráfico, enquanto que, o consumo do modelo 13 aparenta ser o mesmo resultado com filtro, não apresentado o mesmo tipo de curva.

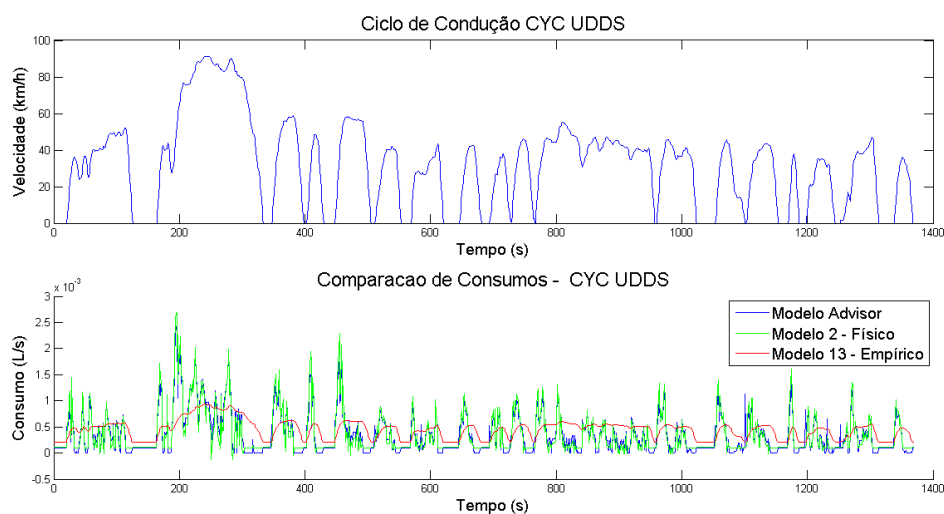


Figura 3.15: Exemplo de consumo dos modelos 2 e 13

3.2.3 Resultados da Calibração

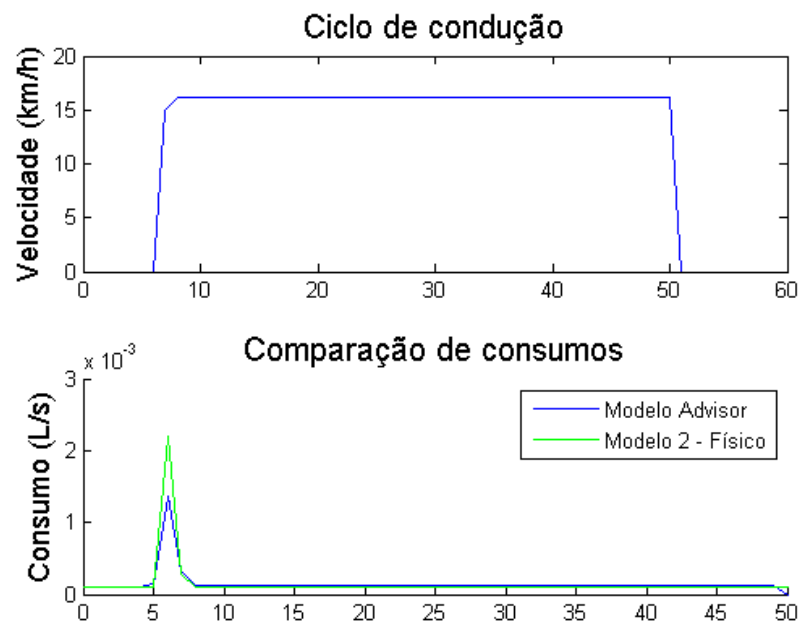


Figura 3.16: Ciclo de calibração 1

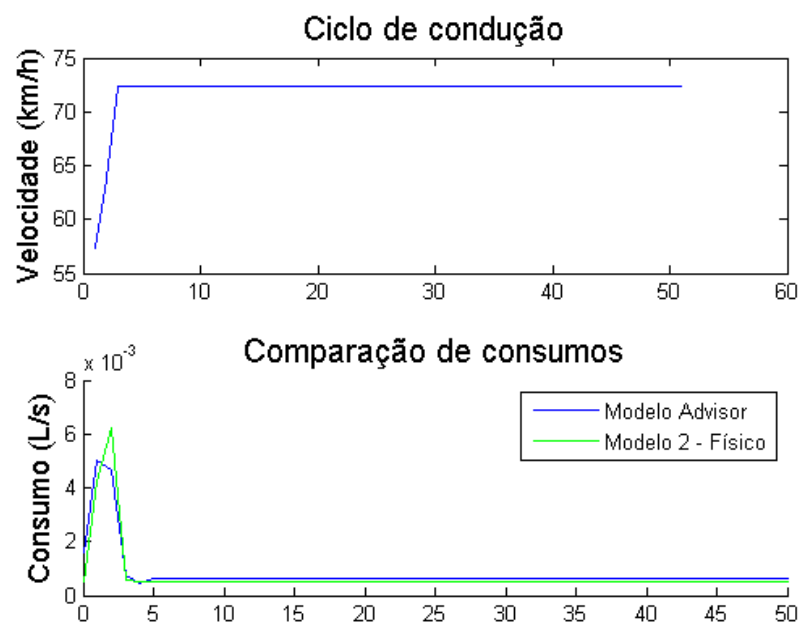


Figura 3.17: Ciclo de calibração 2

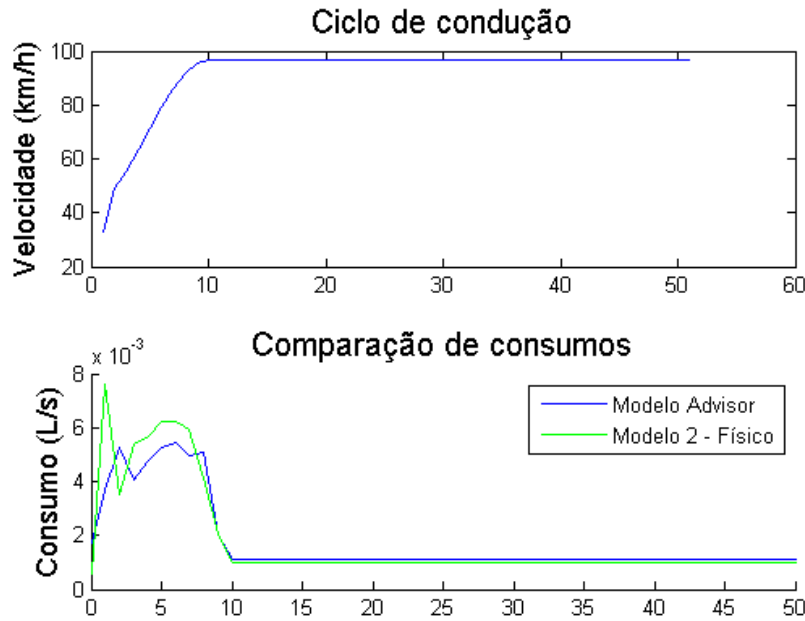


Figura 3.18: Ciclo de calibração 3

As figuras anteriores demonstram o resultado final da calibração. Através das mesmas, é possível observar que o consumo em velocidades constantes é ligeiramente inferior ao consumo do Advisor. Contudo, quando existem acelerações, constata-se que o consumo do Advisor é inferior ao do modelo 2. Considere-se que se optou pela utilização destes parâmetros para que o consumo extra nas acelerações seja compensado nos percursos de velocidade constante.

Com base nestes resultados deu-se por terminada a calibração dos modelos, decidindo-se que o modelo a utilizar seria o 2.

3.3 Sensibilidade do Modelo Físico

Tendo por objetivo compreender de que forma a modificação das entradas provoca variações na estimativa do consumo, escolheu-se utilizar o conceito de sensibilidade. Defina-se a sensibilidade de uma função de acordo com [28], assim:

$$S_{x_i}^{f(x_1, x_2, \dots)} = \frac{\partial f(x_1, x_2, \dots)}{\partial x_i} \frac{x_i}{f(x_1, x_2, \dots)}$$

Considerando esta definição, caso $S_{(x_i)}^{f(x_1, x_2, \dots)} = l$, então se x_i aumentar k então $f(x_1, x_2, \dots)$ aumentará com um fator de k^l .

Neste modelo presente em [10, 13] irá ser analisada a sensibilidade da função de consumo em relação às três entradas do modelo (a, v, θ) . Assim, serão definidas as seguintes sensibilidades: $S_a^{f(a, v, \theta)}$, $S_v^{f(a, v, \theta)}$, $S_\theta^{f(a, v, \theta)}$.

É importante notar que, nestas definições, considerar-se-á que os valores dos parâmetros serão constantes. Sublinhe-se, ainda, que δ será sempre considerado igual a 1, caso contrário as sensibilidades teriam o valor de 0, pelo que seria de pouco relevo calcular tal equação.

3.3.1 Sensibilidade em Função da Aceleração

Defina-se a sensibilidade do consumo em função da aceleração:

$$S_a^{ft(a,v,\theta)} = \frac{\partial ft(a,v,\theta)}{\partial a} \frac{a}{f(a,v,\theta)} \quad (3.1)$$

$$S_a^{ft(a,v,\theta)} = \frac{2\delta(m+m_j)va}{AC_d\delta\rho v^3 + 2C_r\delta gm \cos \theta + 2ft_{idle}\eta_i\epsilon H_g + 2\delta gm \sin \theta + 2a\delta mv + 2a\delta m_jv} \quad (3.2)$$

Substituindo na equação anterior os valores dos parâmetros obtém-se:

$$S_a^{ft(a,v,\theta)} = \frac{42000000va}{16415v^3 + 42000000av + 3531600 \cos \theta + 392400000 \sin \theta + 33010752} \quad (3.3)$$

Atendendo a que a sensibilidade depende de três variáveis, decidiu-se apresentá-la como uma superfície dependente da velocidade e da aceleração para diferentes inclinações. Por esta razão, apresentam-se as seguintes figuras ilustrativas:

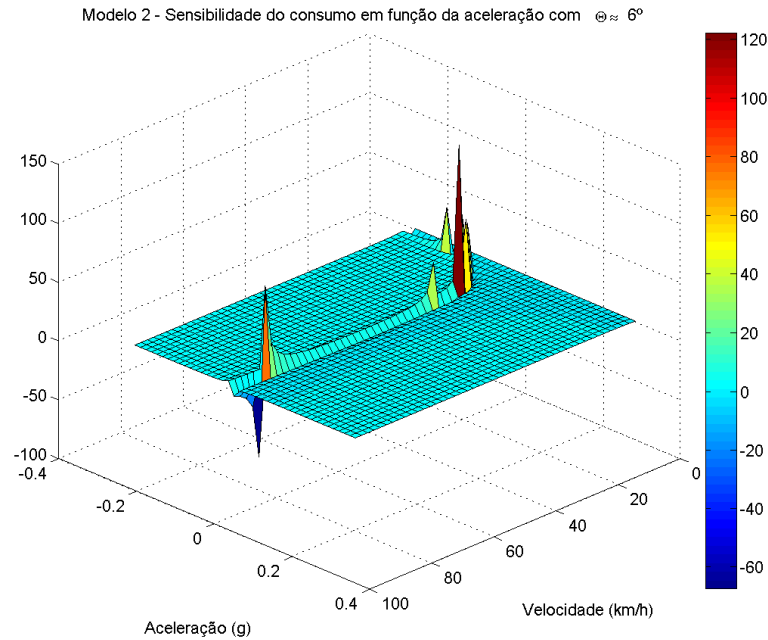


Figura 3.19: Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 6^\circ$

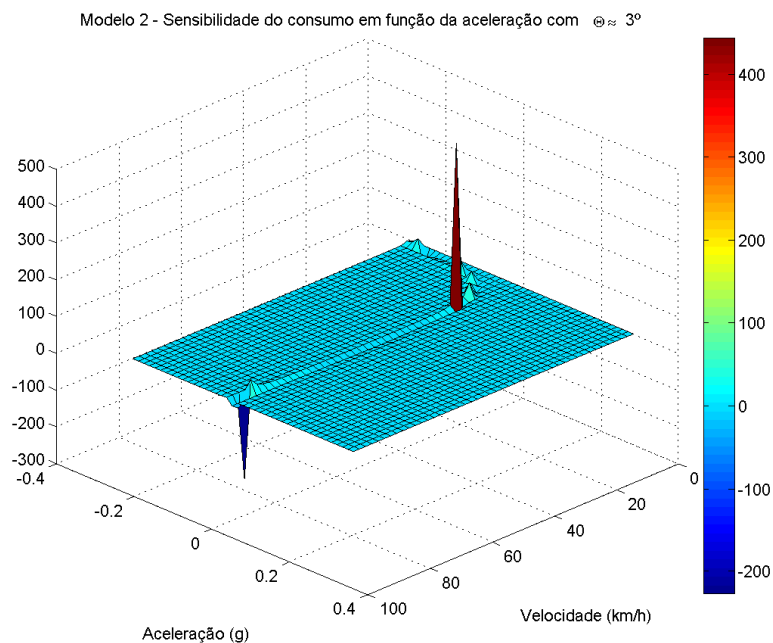


Figura 3.20: Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 3^\circ$

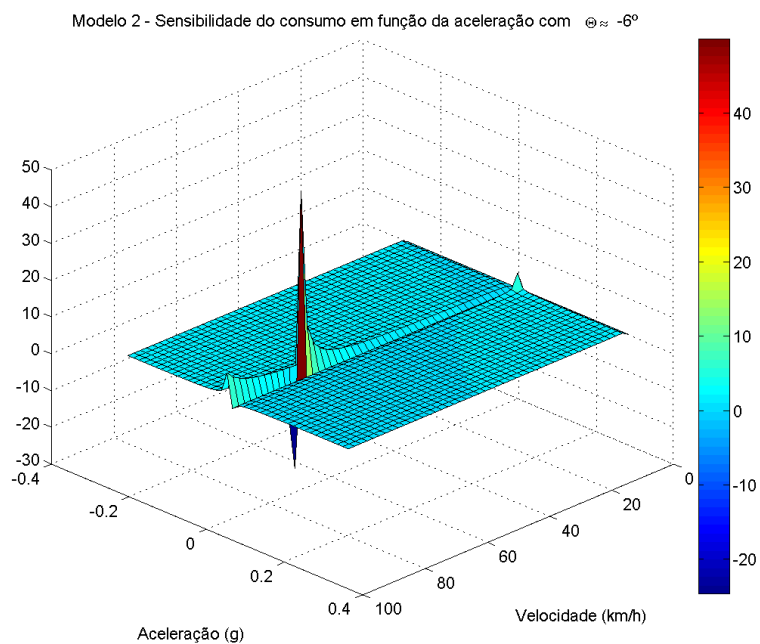


Figura 3.21: Modelo 2 - Sensibilidade do consumo em função de a com $\theta = -6^\circ$

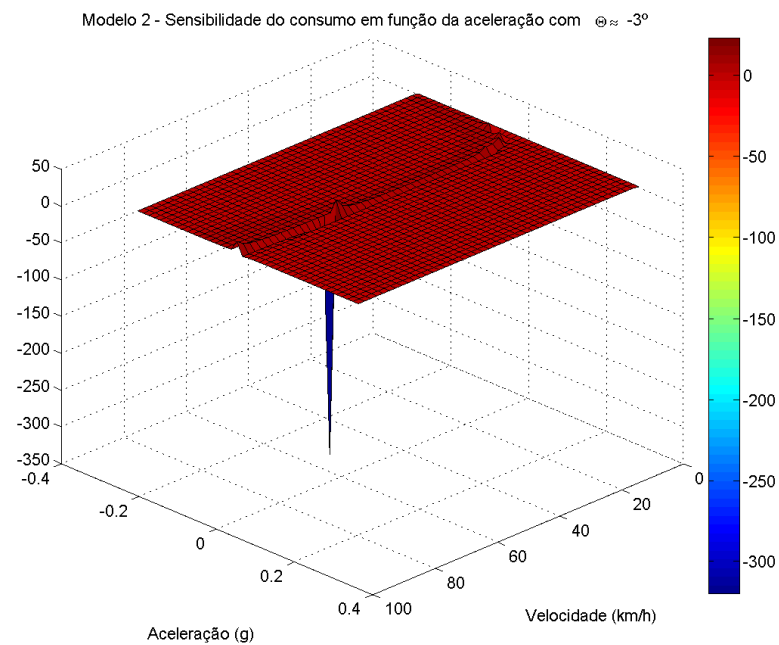


Figura 3.22: Modelo 2 - Sensibilidade do consumo em função de a com $\theta = -3^\circ$

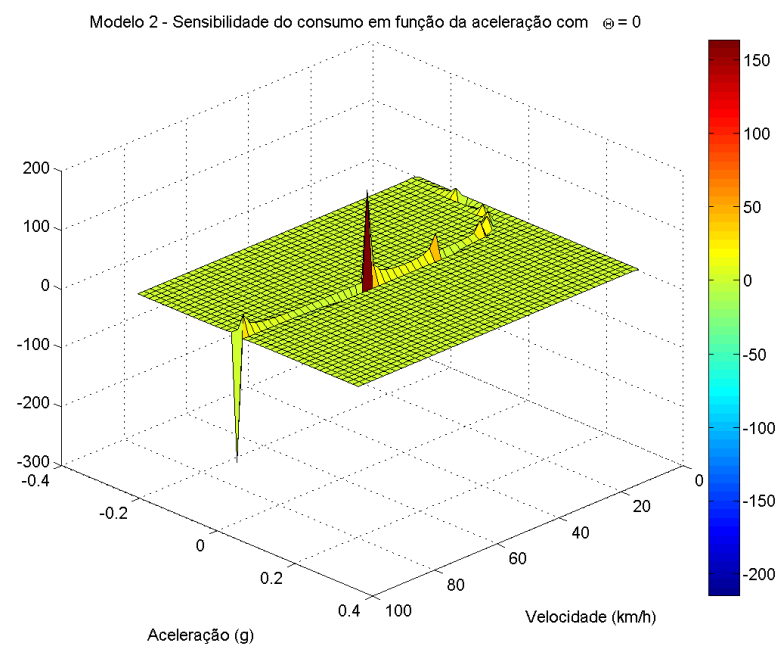


Figura 3.23: Modelo 2 - Sensibilidade do consumo em função de a com $\theta = 0^\circ$

A partir destas figuras percebe-se que a sensibilidade do consumo, de acordo com a variação da aceleração, é, de uma forma geral, constante e próxima de zero. Porém, importa referir que para aceleração nula ou muito próxima de nula, bem como para velocidades baixas, a sensibilidade

apresenta grandes variações, podendo-se observar que variações na aceleração nestas faixas irão provocar grandes variações no consumo.

Para uma melhor avaliação dos valores, apresentam-se, de seguida, alguns valores de sensibilidade para valores típicos de aceleração e de velocidade:

Tabela 3.7: Valores da sensibilidade de a para valores típicos de a e v

$\theta = 0$	$a = -0.2g$	$a = 0g$	$a = 0.2g$
$v = 20km/h$	-25.76	0	0.49
$v = 50km/h$	-0.18	0	0.13
$v = 90km/h$	-0.05	0	0.05

3.3.2 Sensibilidade em Função da Velocidade

Defina-se a sensibilidade do consumo em função da velocidade:

$$S_v^{ft(a,v,\theta)} = \frac{\partial ft(a,v,\theta)}{\partial v} \frac{v}{f(a,v,\theta)} \quad (3.4)$$

$$S_v^{ft(a,v,\theta)} = \frac{\delta(3AC_d\rho v^2 + 2am + 2am_j) * v}{AC_d\delta\rho v^3 + 2C_r\delta gm \cos \theta + 2ft_{idle}\eta_t\epsilon H_g + 2\delta gm \sin \theta + 2a\delta mv + 2a\delta m_j v} \quad (3.5)$$

$$S_v^{ft(a,v,\theta)} = \frac{105v(469v^2 + 400000a)}{16415v^3 + 42000000av + 3531600 \cos \theta + 392400000 \sin \theta + 33010752} \quad (3.6)$$

Realizando uma análise semelhante à efetuada na secção 3.3.1 obtêm-se as seguintes figuras:

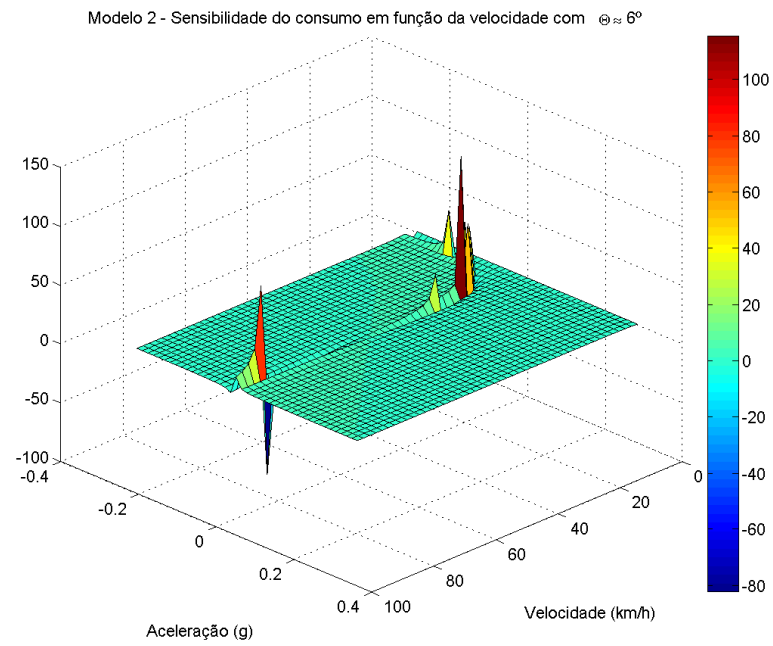


Figura 3.24: Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 6^\circ$

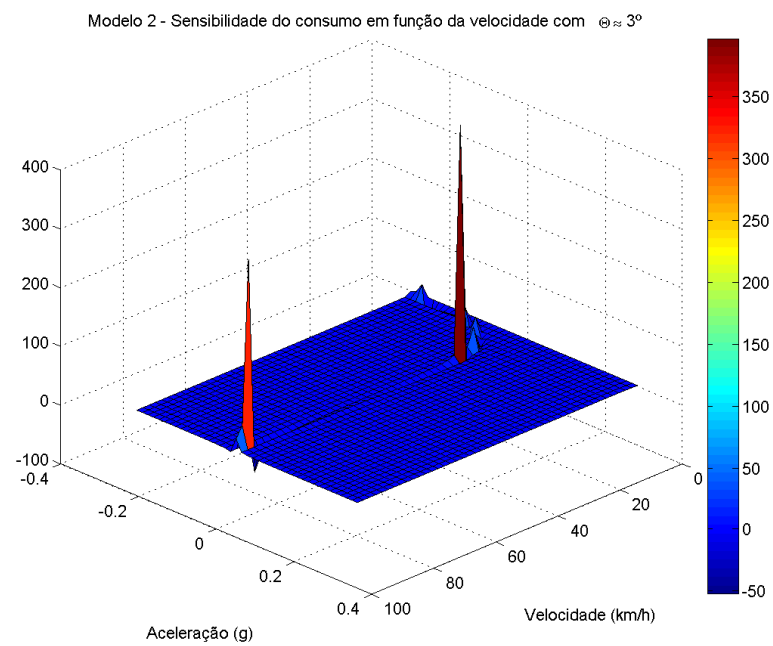


Figura 3.25: Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 3^\circ$

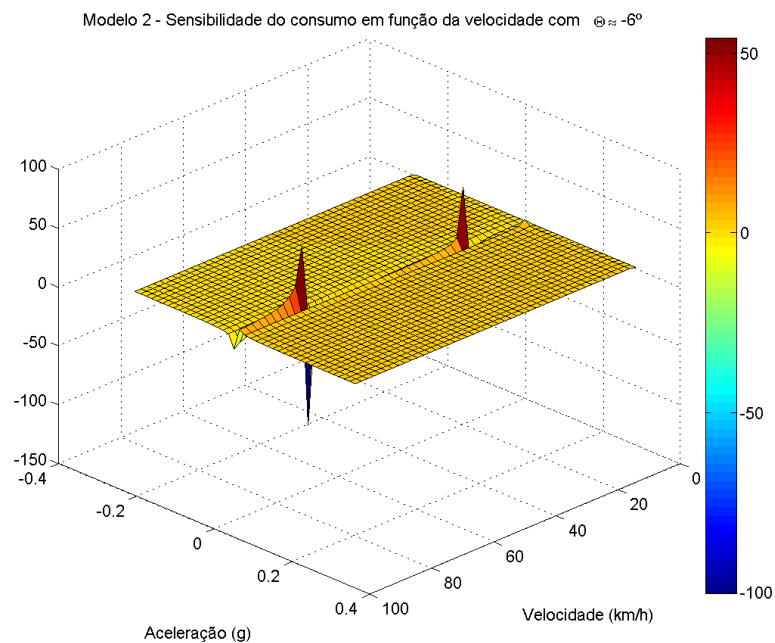


Figura 3.26: Modelo 2 - Sensibilidade do consumo em função de v com $\theta = -6^\circ$

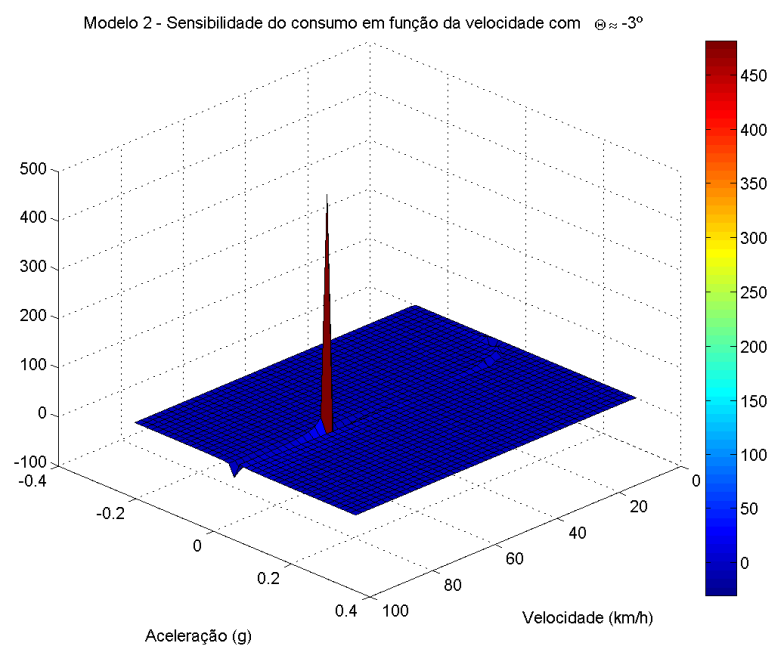


Figura 3.27: Modelo 2 - Sensibilidade do consumo em função de v com $\theta = -3^\circ$

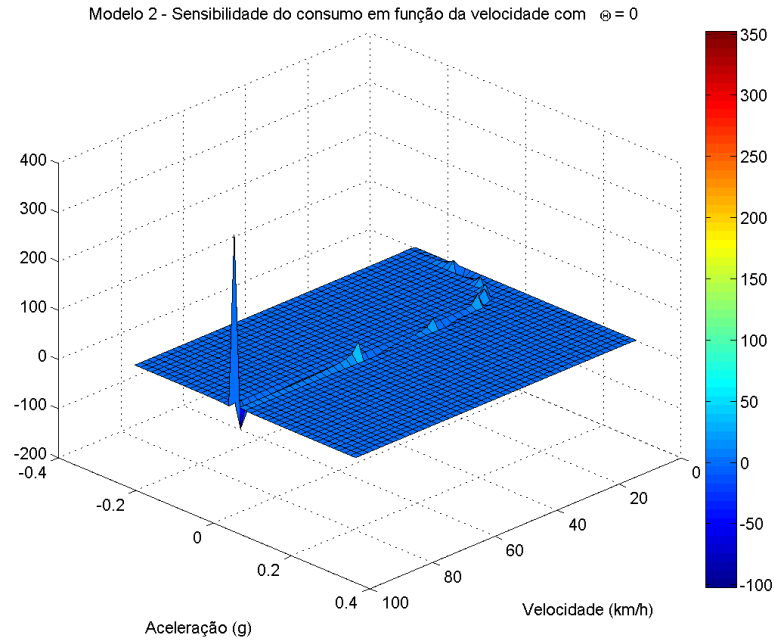


Figura 3.28: Modelo 2 - Sensibilidade do consumo em função de v com $\theta = 0^\circ$

A partir das figuras anteriores, é possível verificar, que, tal como a sensibilidade do consumo em função da aceleração, o mesmo fenómeno se pode notar para a sensibilidade em função da velocidade. Atente-se, contudo, à existência de um pequeno aumento do valor da sensibilidade, o que pode ser ilustrado na seguinte tabela.

Tabela 3.8: Valores da sensibilidade de v para valores típicos de a e v

$\theta = 0$	$a = -0.2g$	$a = 0g$	$a = 0.2g$
$v = 20km/h$	54.04	2.98	2.01
$v = 50km/h$	3.37	3.00	2.73
$v = 90km/h$	3.10	3.00	2.91

3.3.3 Sensibilidade em Função da Inclinação

Defina-se a sensibilidade do consumo em função da inclinação:

$$S_{\theta}^{ft(a,v,\theta)} = \frac{\partial ft(a,v,\theta)}{\partial \theta} \frac{\theta}{f(a,v,\theta)} \quad (3.7)$$

$$S_{\theta}^{ft(a,v,\theta)} = \frac{2\delta mg\theta(C_r \sin \theta - \cos \theta)}{AC_a \delta \rho v^3 + 2C_r \delta gm \cos \theta + 2ft_{idle} \eta_i \varepsilon H_g + 2\delta gm \sin \theta + 2a\delta mv + 2a\delta m_j v} \quad (3.8)$$

$$S_{\theta}^{ft(a,v,\theta)} = \frac{392400\theta(1000\cos\theta - 9\sin\theta)}{16415v^3 + 42000000av + 3531600\cos\theta + 392400000\sin\theta + 33010752} \quad (3.9)$$

Realizando uma análise semelhante às anteriores, obtêm-se as seguintes figuras:

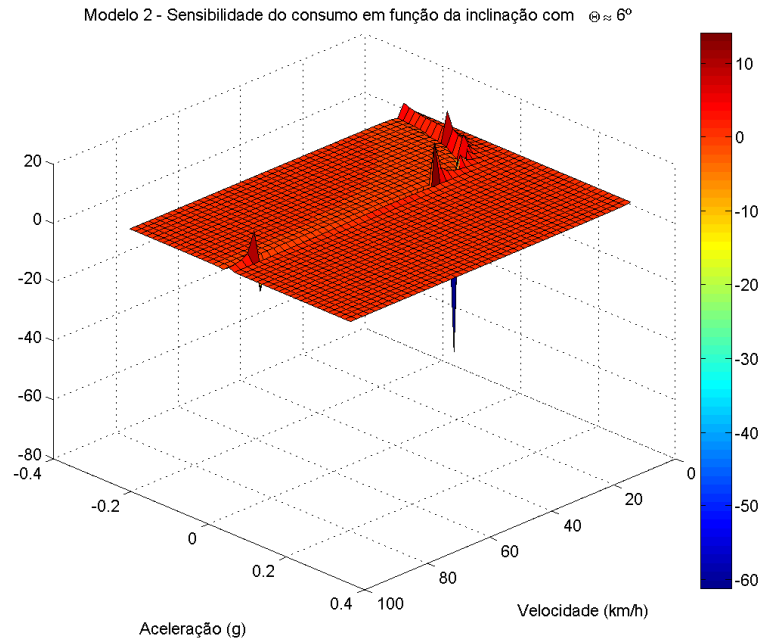


Figura 3.29: Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 6^\circ$

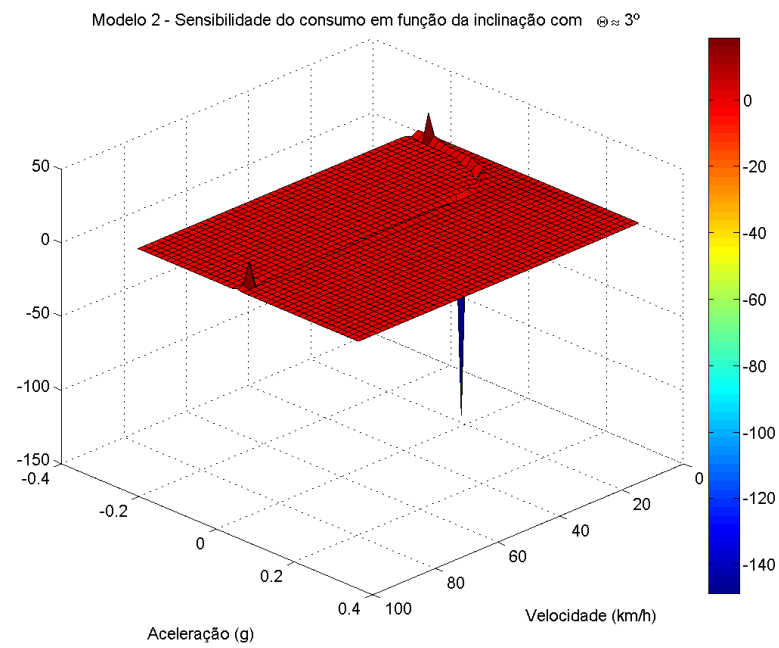


Figura 3.30: Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 3^\circ$

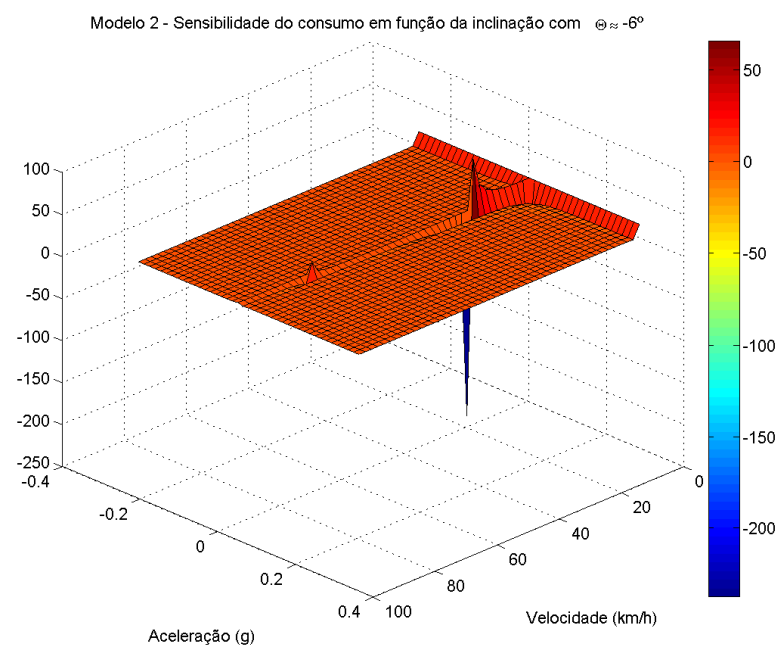


Figura 3.31: Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = -6^\circ$

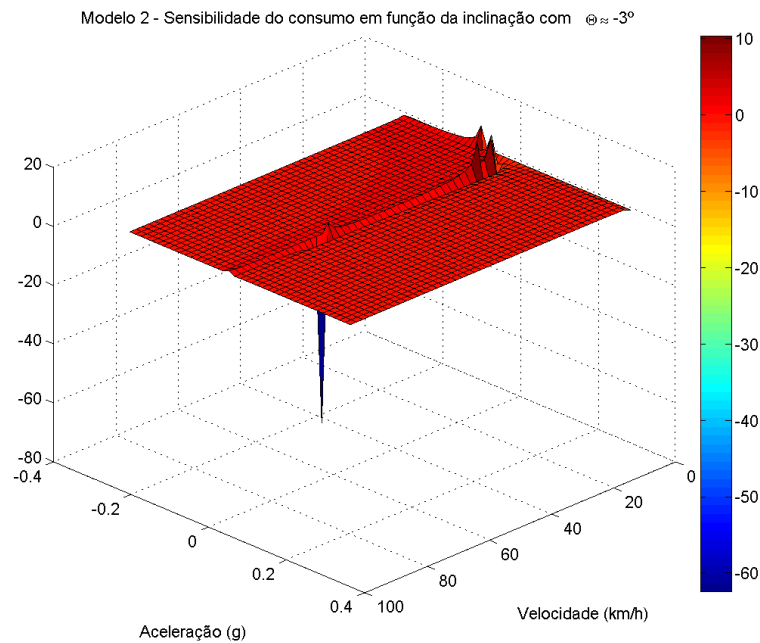


Figura 3.32: Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = -3^\circ$

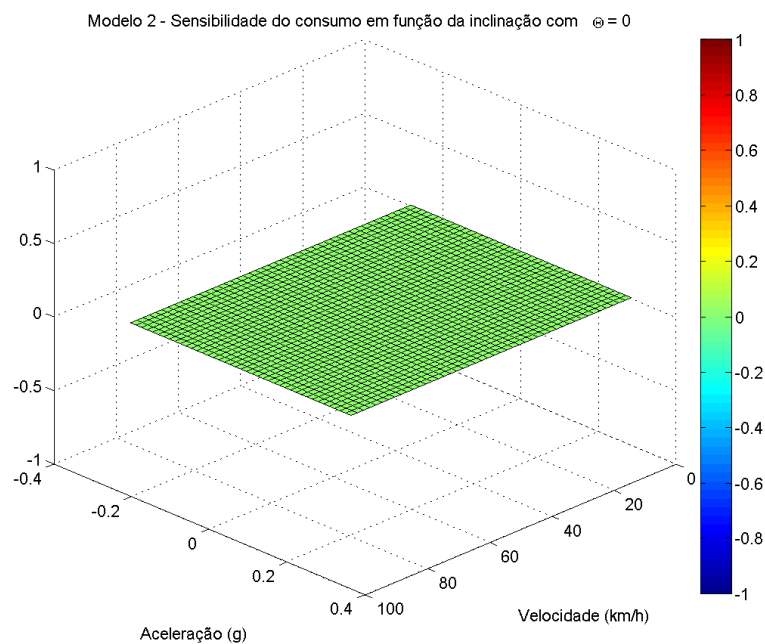


Figura 3.33: Modelo 2 - Sensibilidade do consumo em função de θ com $\theta = 0^\circ$

Pela leitura das figuras acima, obtêm-se conclusões similares às anteriores. Neste caso não se apresenta a sensibilidade para $\theta = 0$, pois com esse valor a sensibilidade será sempre nula. Porém, apresenta-se uma tabela com alguns valores indicativos para a sensibilidade para $\theta = 3^\circ$.

Tabela 3.9: Valores da sensibilidade de v para valores típicos de a e v

$\theta = 0.05$	$= -0.2g$	$a = 0g$	$a = 0.2g$
$v = 20km/h$	0.07	3.2×10^{-3}	1.6×10^{-3}
$v = 50km/h$	2.41×10^{-4}	2.04×10^{-4}	1.77×10^{-4}
$v = 90km/h$	3.68×10^{-5}	3.51×10^{-5}	3.35×10^{-5}

Pela análise destes resultados pode-se notar que a sensibilidade é quase zero ao longo de grande parte da superfície.

3.3.4 Conclusões

Com base nos resultados anteriores, é possível perceber que a sensibilidade é bastante constante ao longo das várias acelerações e velocidades, à exceção de algumas zonas próximas da aceleração nula.

3.4 Simulação do Modelo Físico

Tendo em linha de conta a calibração realizada na secção anterior, executaram-se simulações para vários ciclos de condução que foram extraídos da plataforma Advisor. Ao contrário dos ciclos usados na calibração, os agora simulados permitem perceber o comportamento do estimador para ciclos de condução mais usuais. Por outro lado, devido às características da aquisição de sinal pretendida, considerou-se a utilização de períodos de amostragem variáveis. Teve-se, ainda, em atenção que estes dispositivos poderão ser influenciados por ruído, pelo que o mesmo foi incluído nas entradas. Por último, considerou-se o uso de vários métodos de derivadas.

3.4.1 Efeitos de Diferentes Períodos de Amostragem

Nesta secção serão apresentados os vários ciclos de condução e os respetivos consumos. Pretende-se, aqui, avaliar a influência do período de amostragem (T_s) na estimativa do consumo de combustível, tendo em vista inferir conclusões para possíveis problemas na implementação do algoritmo no dispositivo móvel.

3.4.1.1 Ciclos de velocidade constante

Para uma análise preliminar, apresenta-se os resultados de simulações para ciclos a velocidade constante, contendo, apenas, uma aceleração inicial.

- CYC CONSTANT

Neste ciclo é possível visualizar que o desempenho do estimador piora com o aumento do período de amostragem. Contudo é importante notar que o modelo continua a responder corretamente para velocidades constantes. Importa, ainda, realçar que o aumento do T_s leva a que grandes acelerações não sejam analisadas pelo estimador.

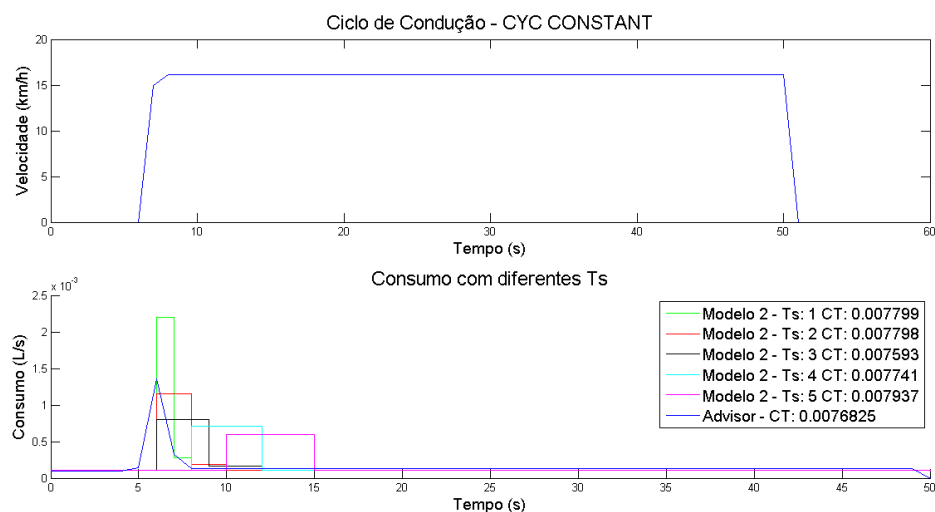


Figura 3.34: Comparação de consumo com diferentes T_s para o ciclo: CYC CONSTANT

- CYC CONSTANT 60

Nesta simulação é possível observar que a estimativa do modelo 2 é sempre maior ao consumo obtido do Advisor, tal poderá dever-se à elevada aceleração e aos problemas que ela provoca na estimativa do consumo. Nota-se, ainda, que com o aumentar do T_s a estimativa piora.

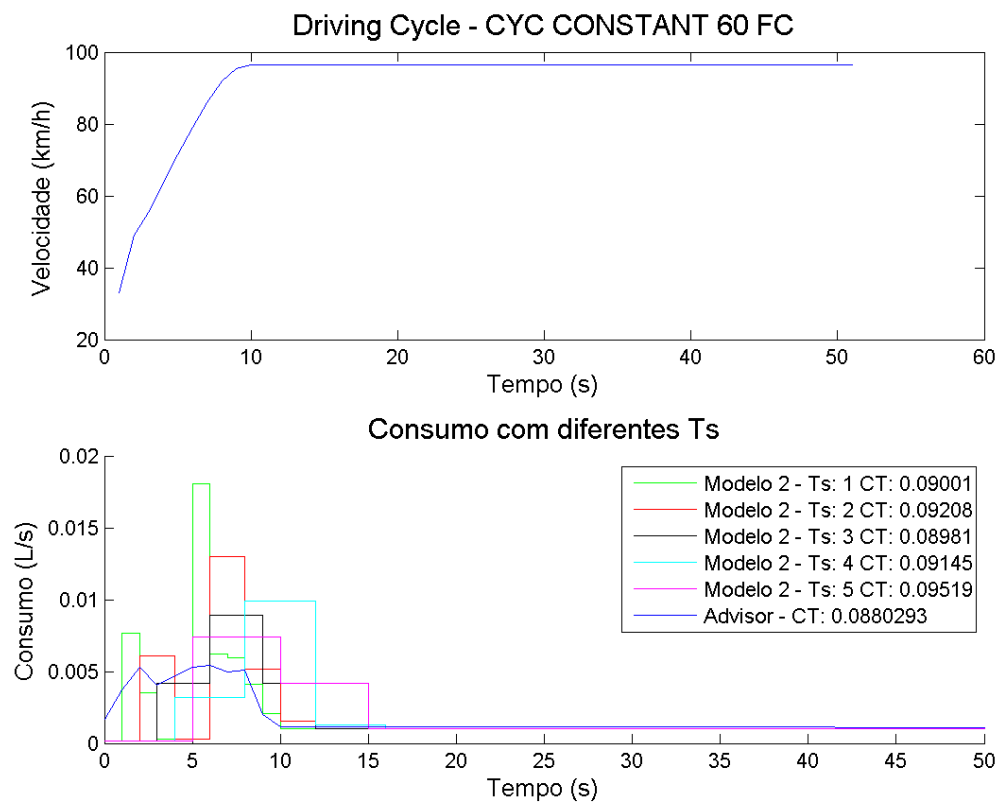


Figura 3.35: Comparação de consumo com diferentes Ts para o ciclo: CYC CONSTANT 60

3.4.1.2 Ciclos de velocidade variável

Nesta secção pretende-se verificar de que forma os diferentes períodos de amostragem provocam alterações em ciclos de condução mais complexos e próximos da realidade.

- CYC CLEVELAND

Neste primeiro ciclo existem elevadas acelerações, o que provoca um grande erro na estimativa do consumo. Curiosamente, quanto maior o Ts menor o erro de estimativa. Tal poderá dever-se ao facto das elevadas acelerações passarem a ser "ignoradas".

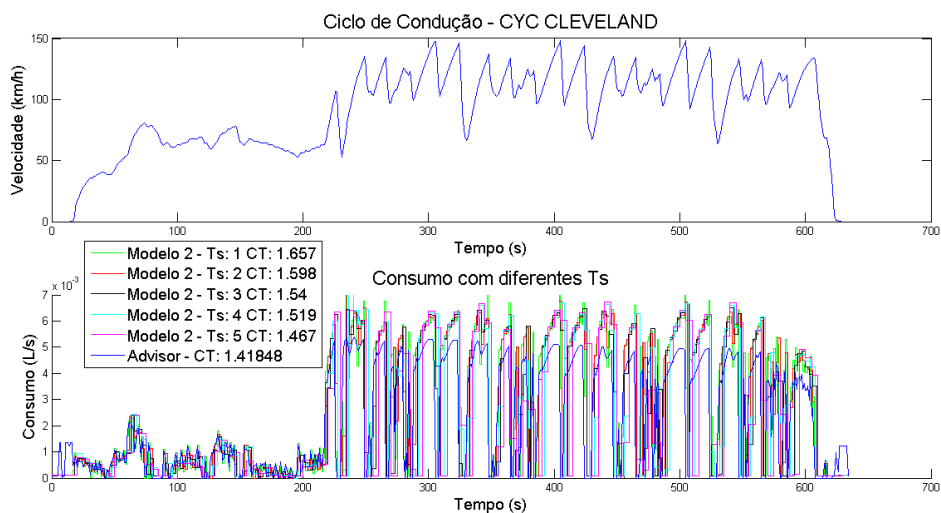


Figura 3.36: Comparação de consumo com diferentes Ts para o ciclo: CYC CLEVELAND

- CYC HWFET

Neste segundo ciclo é possível observar-se que o consumo estimado pelo modelo 2 é sempre inferior ao do obtido do Advisor, tal dever-se-á ao facto de a velocidade ser quase constante e a aceleração ser muito reduzida.

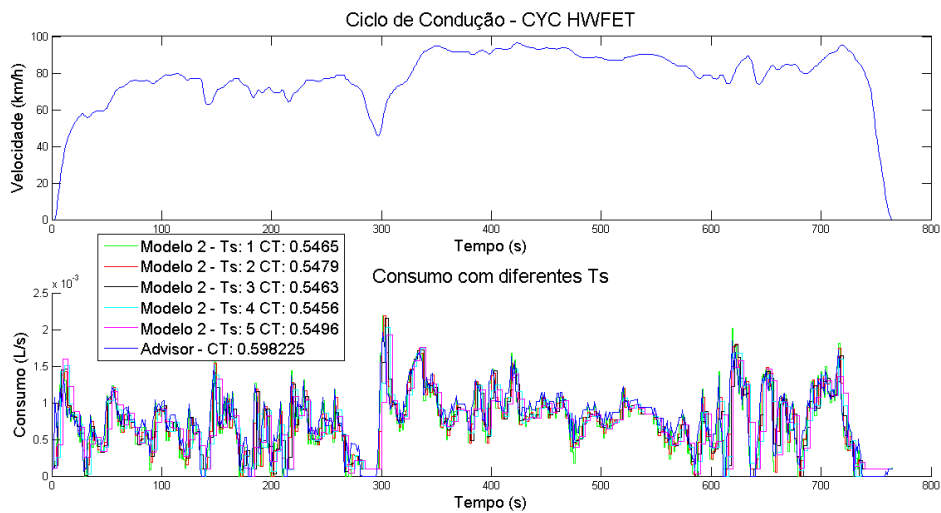


Figura 3.37: Comparação de consumo com diferentes Ts para o ciclo: CYC HWFET

- CYC MANHATTAN

No terceiro ciclo de condução, podem-se observar grandes acelerações em gamas de velocidade baixas. É um típico ciclo citadino. Neste é possível constatar que a estimativa do modelo 2 é superior à apresentada pelo Advisor para todos os Ts, piorando com o aumento do Ts.

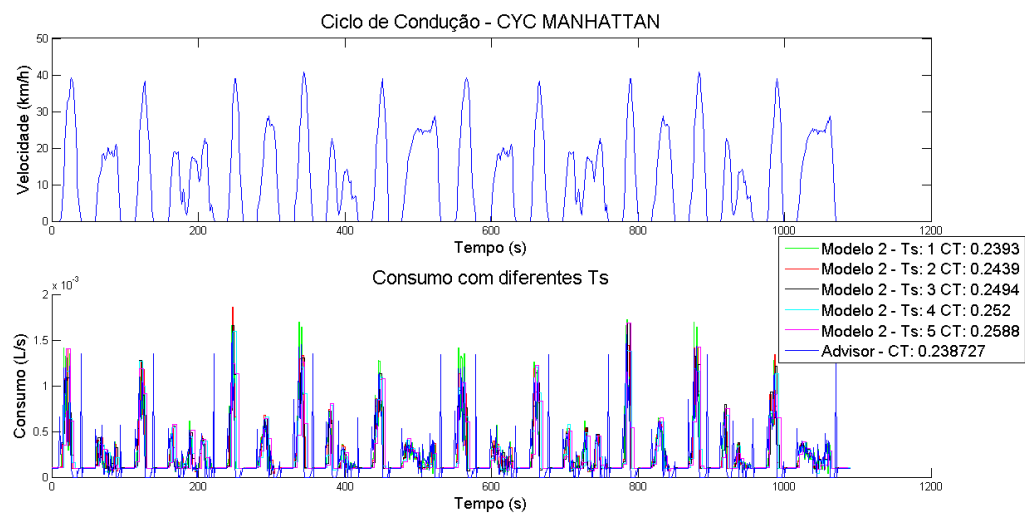


Figura 3.38: Comparação de consumo com diferentes Ts para o ciclo: CYC MANHATTAN

- CYC UDDS

Trata-se de um ciclo bastante complexo, permitindo visualizar praticamente todos os tipos de variações possíveis. Tendo tal em consideração, pode-se concluir que é um excelente ciclo de condução para comparação, prevendo-se que com o aumento do Ts as estimativas irão piorar.

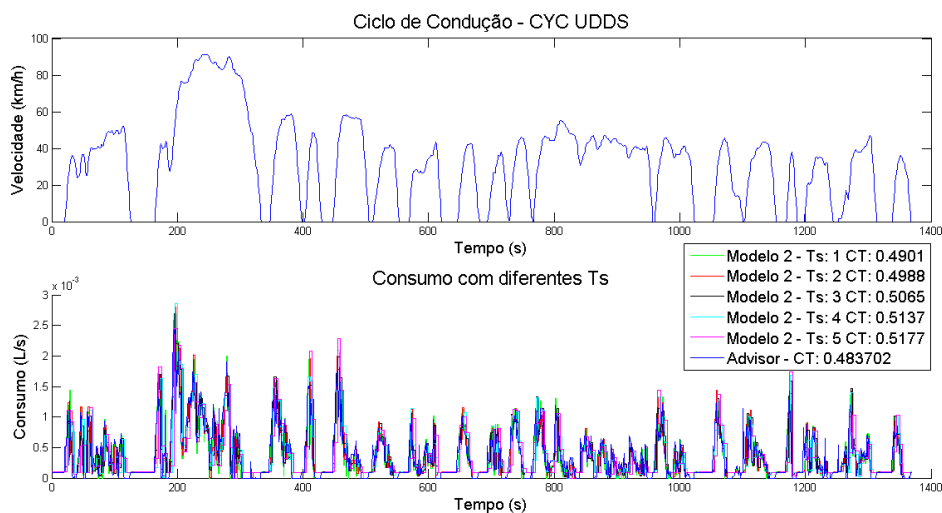


Figura 3.39: Comparação de consumo com diferentes Ts para o ciclo: CYC UDDS

Em resumo, esta análise permite-nos avaliar a estimativa do consumo do modelo 2 para vários períodos de amostragem. Assim, é possível concluir que, na generalidade, o aumento do período de amostragem piora a estimativa do consumo, excetuando os casos em que existem acelerações bastante elevadas.

3.4.2 Impacto do Ruído de Aquisição nas Estimativas

Pretende-se, neste ponto, realizar uma análise semelhante à da secção 3.4.1, sendo que a diferença será a introdução de ruído branco (média nula) nas entradas, de forma a simular os erros de aquisição dos sensores do dispositivo móvel. No caso das figuras abaixo, o ruído branco terá como desvio padrão (σ) $\sqrt{2}$ km/h.

3.4.2.1 Ciclos de velocidade constante

- CYC CONSTANT

Neste primeiro ciclo é possível verificar que a performance do estimador é bastante afetada pelo ruído branco para todos os Ts, afetando especialmente quando o período de amostragem é menor.

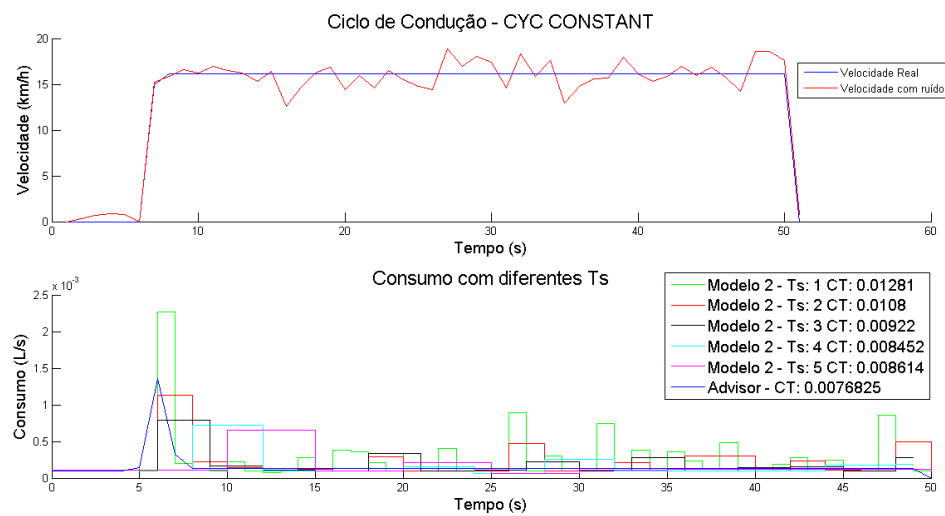


Figura 3.40: Comparação de consumo, introduzindo ruído para o ciclo: CYC CONSTANT

• CYC CONSTANT 60

Na presente figura observa-se o efeito já observado anteriormente. Todavia, pode verificar-se que, devido às velocidades mais elevadas, o consumo não é tão afetado.

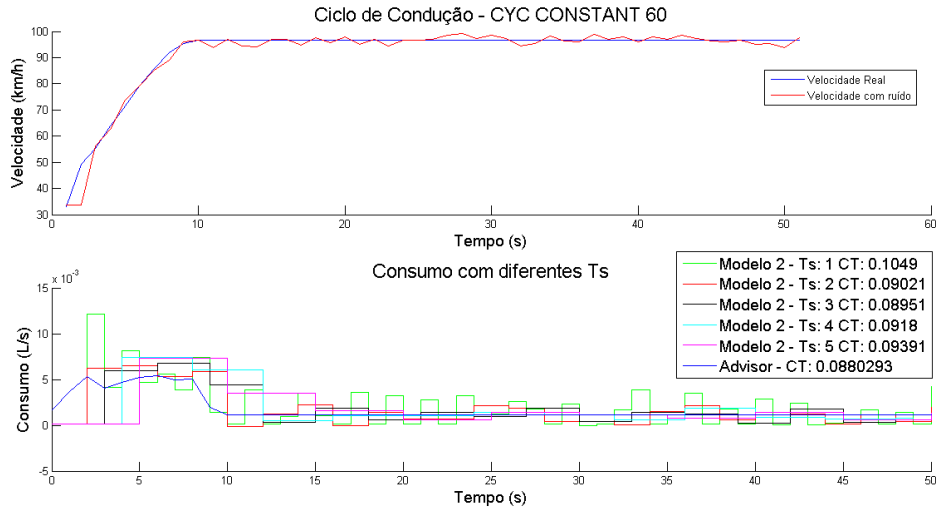


Figura 3.41: Comparação de consumo, introduzindo ruído para o ciclo: CYC CONSTANT 60

3.4.2.2 Ciclos de velocidade variável

- CYC CLEVELAND

Neste terceiro ciclo, de forma análoga, é possível observar que o erro parece ser maior para T_s menores. Tal poderá dever-se ao mesmo facto de as elevadas acelerações, provocadas pela introdução de ruído e pelo padrão do ciclo, passarem a ser "ignoradas".

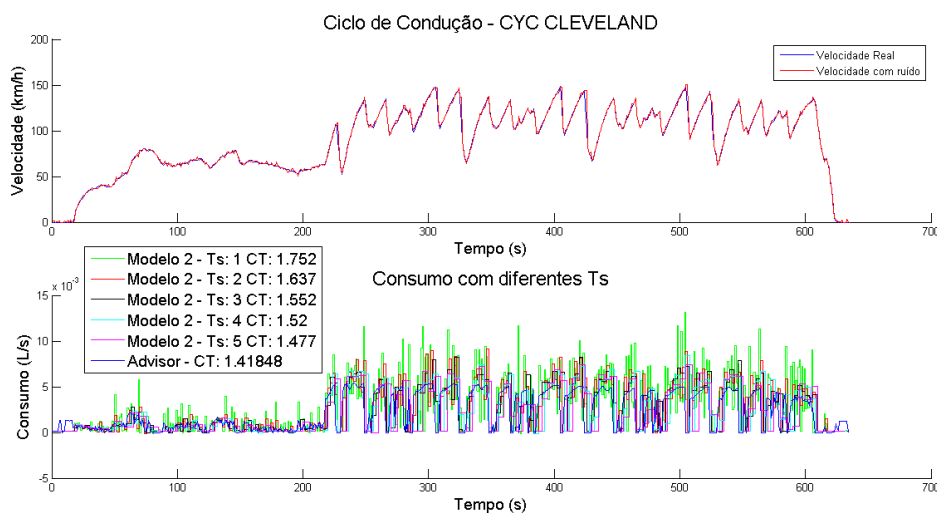


Figura 3.42: Comparação de consumo, introduzindo ruído para o ciclo: CYC CLEVELAND

- CYC HWFET

Neste quarto ciclo é possível observar que o consumo estimado tem um comportamento aproximado ao dos anteriores. Contudo, ao contrário dos daqueles, não se aproxima mais do valor real para o período de amostragem máximo.

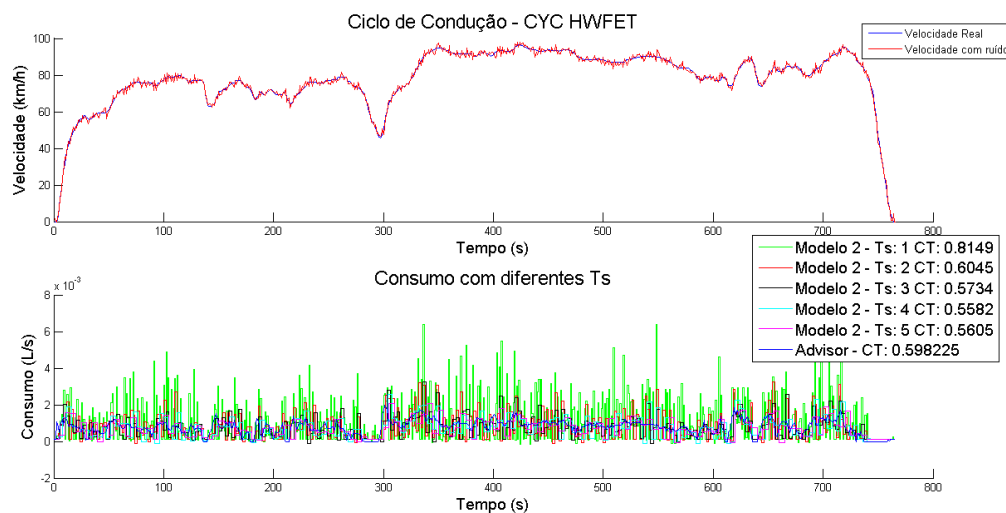


Figura 3.43: Comparação de consumo, introduzindo ruído para o ciclo: CYC HWFET

- CYC MANHATTAN

Neste quinto ciclo de condução citadino pode-se observar que o efeito do ruído é sentido de forma semelhante para todos os Ts, nunca se aproximando, realmente, do valor real. Isto pode dever-se ao facto de, para Ts baixos, as acelerações serem mais acentuadas e, para Ts altos, serem "ignoradas".

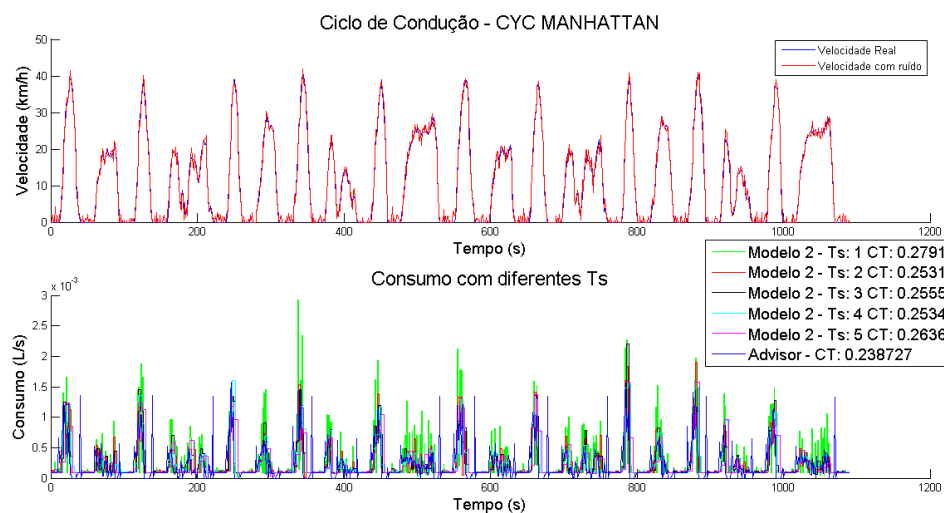


Figura 3.44: Comparação de consumo, introduzindo ruído para o ciclo: CYC MANHATTAN

- CYC UDDS

No sexto ciclo é possível verificar o padrão no qual, com o aumentar do período de amostragem, o estimador se aproxima da realidade.

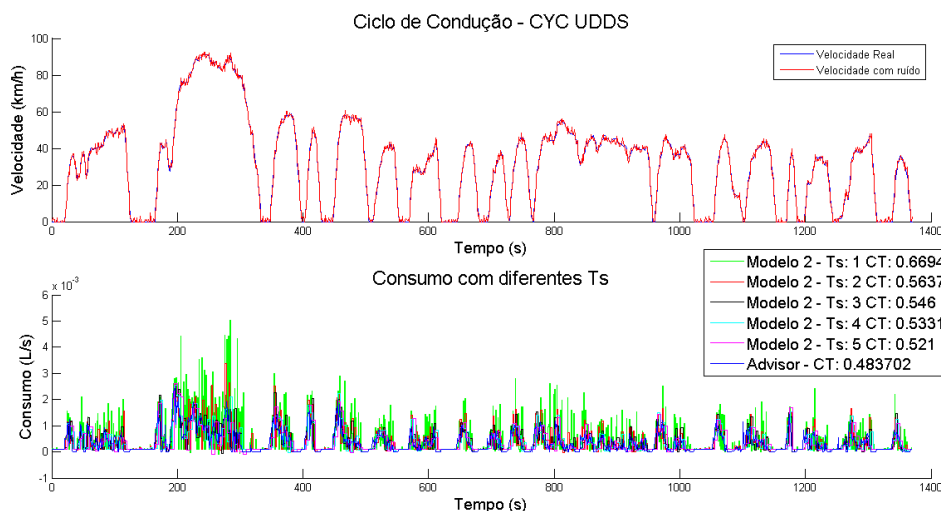


Figura 3.45: Comparação de consumo, introduzindo ruído para o ciclo: CYC UDDS

Em suma, é possível concluir que a aceleração afeta bastante a estimativa do consumo de combustível, podendo ser necessário a introdução de algumas correções no estimador. É, também, relevante referir que, com o aumento do período de amostragem, parece existir uma diminuição do erro de estimação. Tal dever-se-á, em parte, ao facto das acelerações serem atenuadas nos percursos de velocidade aproximadamente constante. Contudo, é necessário ponderar a utilização de períodos de amostragem elevados, visto poderem levar a situações em que acelerações/desacelerações rápidas são ignoradas.

De referir, ainda, que a introdução de ruído provoca um maior erro para velocidades mais reduzidas.

3.4.3 Estudo das Influências dos Métodos de Cálculo da Aceleração

A presente secção detém-se sobre o problema da derivação da velocidade para aquisição da aceleração. Esta questão surge da dificuldade de obtenção de leituras corretas dos acelerómetros dos *smartphones*, os quais apresentaram grandes erros nos percursos de teste realizados. Como método de resolução deste problema considerou-se, então, a derivação da velocidade. Não obstante, encontrámo-nos no domínio discreto, pelo que é necessário calcular a derivada utilizando métodos numéricos. Assim, recorreu-se a [29], onde são apresentados vários métodos que serão seguidamente expostos:

3.4.3.1 Métodos de derivação

Os métodos de derivação apresentados em [29] são condensados na seguinte expressão:

$$f^{(k)}(x_j) = \frac{k!}{n!} \frac{1}{h^k} \sum_{i=0}^n a_i f(x_i) \quad (3.10)$$

onde k é a ordem da derivada, neste caso 1, h é o passo, ou o tempo de aquisição neste caso, i o instante referente da velocidade, onde n é o instante atual, e 0 o máximo instante anterior considerado os coeficientes a são retirados da tabela 3.10, que pode ser encontrada em [29].

Tabela 3.10: Coeficientes de derivadas de primeira ordem

Primeira Derivada						
j	a_0	a_1	a_2	a_3	a_4	erro
0	-1	1				$-\frac{1}{2}hf^{(2)}$
1	-1	1				$\frac{1}{2}hf^{(2)}$
0	-3	4	-1			$\frac{1}{3}h^2f^{(3)}$
1	-1	0	1			$-\frac{1}{6}h^2f^{(3)}$
2	1	-4	3			$\frac{1}{3}h^2f^{(3)}$
0	-11	18	-9	2		$-\frac{1}{4}h^3f^{(4)}$
1	-2	-3	6	1		$\frac{1}{12}h^3f^{(4)}$
2	1	-6	3	2		$-\frac{1}{12}h^3f^{(4)}$
3	-2	9	-18	11		$\frac{1}{4}h^3f^{(4)}$
0	-50	96	-72	32	-6	$\frac{1}{5}h^4f^{(5)}$
1	-6	-20	36	-12	2	$-\frac{1}{20}h^4f^{(5)}$
2	2	-16	0	16	-2	$\frac{1}{30}h^4f^{(5)}$
3	-2	12	-36	20	6	$-\frac{1}{20}h^4f^{(5)}$
4	6	-32	72	-96	50	$\frac{1}{5}h^4f^{(5)}$

Nesta tabela é possível observar que a derivada pode ser calculada recorrendo a vários pontos. Pode ser calculada considerando apenas ou os pontos anteriores, ou os pontos anteriores e posteriores ou apenas posteriores. Tendo em consideração o problema a desenvolver, considerar-se-ão unicamente os casos em que o cálculo recorre apenas aos pontos anteriores, ou seja, os índices a considerar serão sempre os últimos de cada grupo de índices. Esta opção foi tomada visto o tempo de amostragem ser elevado, provocando um maior atraso na apresentação do consumo em tempo real. Realizaram-se algumas experiências, incluindo este atraso, mas os resultados não foram satisfatórios.

3.4.3.2 Avaliação dos métodos

Para verificar o comportamento destes métodos, usaram-se alguns dos ciclos de condução previamente obtidos e, com recurso ao MATLAB/Simulink, realizaram-se as derivadas pretendidas, tendo-se obtido os seguintes resultados:

- Exemplo de derivação nº1

Analisando a figura 3.46 é possível observar que para os métodos 5 e 4 (utilização de 5 e 4 pontos respetivamente), a derivada apresenta valores que não estão de acordo com o pretendido. Na zona realçada, a velocidade encontra-se a aumentar, todavia a derivada apresenta um valor negativo, o que não pode acontecer pois a derivada no ponto, por definição, é o declive da reta tangente à curva, ou seja, teria de ser positiva.

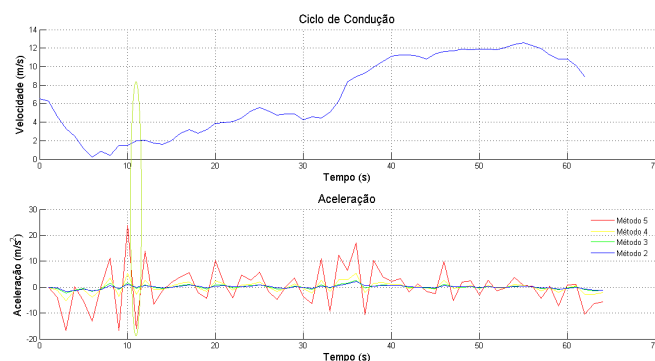


Figura 3.46: Exemplo de derivação numérica 1

- Exemplo de derivação nº2

Na figura 3.47 é possível observar o fenómeno semelhante, mas para o caso inverso. É, ainda, possível notar que é apresentada uma desaceleração de módulo superior a $40m/s^2$. Este valor é bastante irrealista, tendo em consideração a dinâmica de um automóvel.

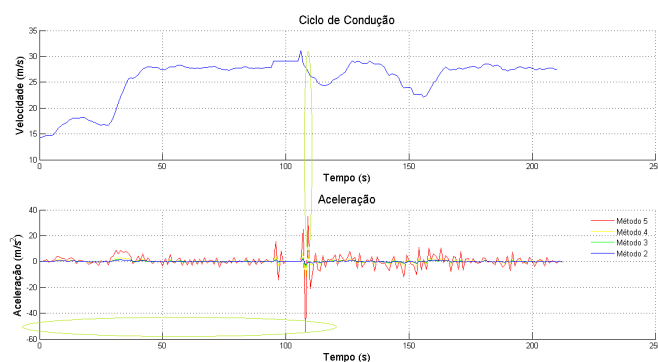


Figura 3.47: Exemplo de derivação numérica 2

- Exemplo de derivação nº3

Nesta figura 3.48 pode observar-se o fenómeno demonstrado anteriormente. Assim, para uma melhor análise, será apresentado o mesmo gráfico 3.49. Não obstante, a velocidade será representada, apenas, nos pontos em que está a ser amostrada, ou seja, poderão ser observados os pontos que estão a ser considerados nas diversas derivações.

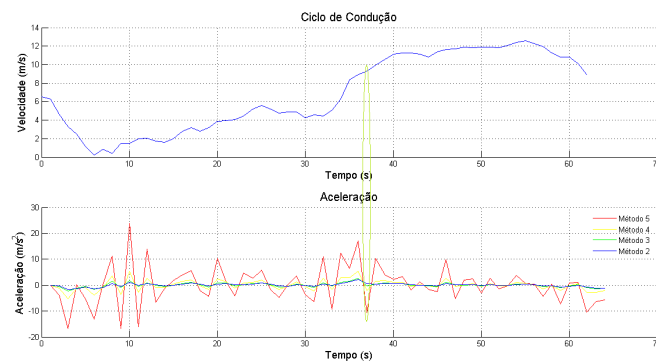


Figura 3.48: Exemplo de derivação numérica 3.1

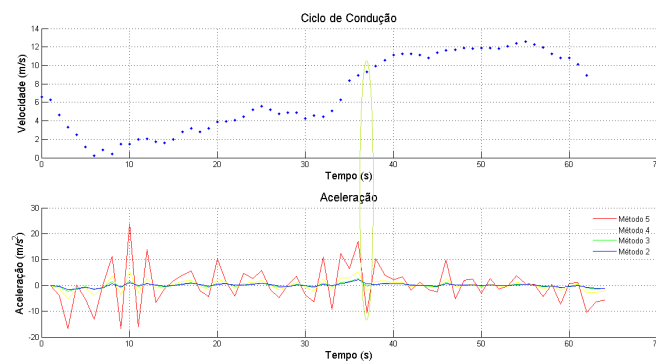


Figura 3.49: Exemplo de derivação numérica 3.2

Através desta figura é possível perceber que, devido a uma aceleração mais rápida entre os pontos 2 e 3, antes do ponto de derivada obtém-se uma derivada errada no ponto.

3.5 Conclusões

Após a calibração e as simulações do modelo, conclui-se que a aceleração do veículo afeta, de forma drástica, o comportamento do estimador de combustível. Sugere-se, assim, a introdução de um fator extra no seguinte elemento da estimativa: $(m + m_j)av$. Isto permitirá reduzir o efeito da aceleração no consumo, melhorando a estimativa do mesmo.

Saliente-se que, para todas as simulações realizadas, foi considerada inclinação nula ao longo de todo o ciclo de condução, ou seja, $\theta = 0$ em todos os instantes, podendo, assim, ser sugerida uma simplificação no modelo, no qual θ tomaria sempre o valor de 0.

Tendo em vista melhorar a compreensão do modelo, propõe-se, igualmente, substituir as duas eficiências presentes no modelo por uma única, que representa a eficiência total do veículo. Esta ação seria tomada tendo em consideração que as duas eficiências se encontram sobre a forma de um produto, no cálculo do consumo de combustível.

Capítulo 4

Formulação do Novo Algoritmo

Com o presente capítulo pretende-se apresentar o algoritmo desenvolvido para melhorar a estimativa do consumo de combustível do modelo físico explicado e analisado no capítulo anterior. Neste sentido, começa-se por descrever o algoritmo auxiliar capaz de detetar o tipo de ciclo que o veículo se encontra a percorrer para, de seguida, se proceder, então, à apresentação do novo algoritmo.

4.1 Algoritmo para Detecção de Tipo de Ciclo de Condução

Nesta secção será descrito o algoritmo desenvolvido para detecção do tipo de ciclo de condução. Pretende-se, com este, distinguir entre os seguintes tipos: *Stop and Go*, Urbano, Semi-Urbano, Via rápida e Auto-estrada, sendo que nos dois últimos se distinguirá, ainda, em função do volume de tráfego (muito ou pouco), visto que o consumo pode diferir de acordo com esta variável.

4.1.1 Entradas e Saídas do Algoritmo

Este algoritmo terá por base o algoritmo apresentado em [30], no qual é utilizado um bloco de lógica difusa, cuja entrada principal é a velocidade média, permitindo a distinção de ciclos de condução em 3 classes. Para melhorar esta distinção de entradas, considerou-se, também, o tempo parado, brevemente referenciado em [30]. Esta entrada terá o principal papel de distinguir entre o tráfego Urbano e *Stop and Go*. Para o cálculo do tempo parado, considere-se que o veículo está imobilizado sempre que a velocidade seja menor ou igual a 5km/h .

Considerando estas duas entradas é possível distinguir o ciclo de condução em cinco classes, sendo, no entanto, impossível distinguir o volume de tráfego. Para isso, recorreu-se à energia da aceleração, tal como definida em [31] e representada pela seguinte equação.

$$E_a[n] = a^2[n-1] - a[n] * a[n-2] \quad (4.1)$$

onde E_a e a representam a energia da aceleração e a aceleração, respectivamente. Esta nova entrada permitirá inferir que, caso a energia da aceleração seja elevada, provavelmente estaremos numa zona de grande intensidade de tráfego, visto ser necessário acelerar e desacelerar com frequência.

De referir ainda que, para todas as entradas, se recorreu a um filtro de média móvel com uma escala temporal de 50 segundos, como sugerido em [32]. Este filtro é combinado com o valor no instante anterior, para que se possa ter uma maior perceção da variação dos valores. O mesmo apresenta a seguinte equação.

$$\bar{x}(t) = \alpha x(t-1) + \frac{1-\alpha}{N} \sum_{i=0}^{N-1} x(t-i) \quad (4.2)$$

No nosso caso, temos que $N = 50$, faltando apenas a definição de α . Após algumas inferências, tomou-se o valor de $\alpha = 0.2$.

Por último, definiu-se uma escala de saída entre 0 e 100, sendo que, para cada intervalo, será atribuído um tipo de ciclo.

4.1.2 Funções de Pertença e Regras de Inferência

Tendo em consideração as entradas definidas anteriormente (velocidade média, tempo parado e energia de aceleração) e o objetivo deste algoritmo, definiram-se as seguintes funções de pertença para as primeiras:

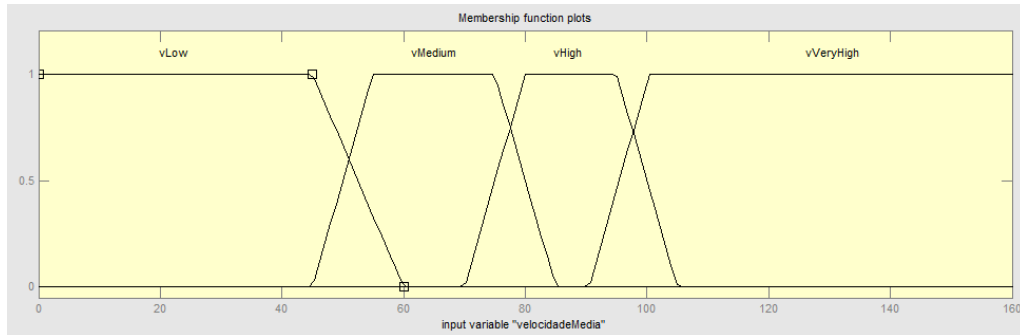


Figura 4.1: Função de pertença da velocidade média

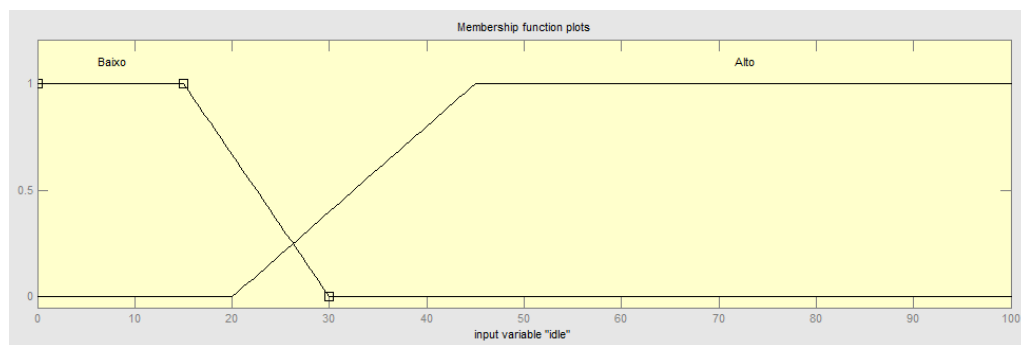


Figura 4.2: Função de pertinência do bloco de lógica difusa para o tempo parado

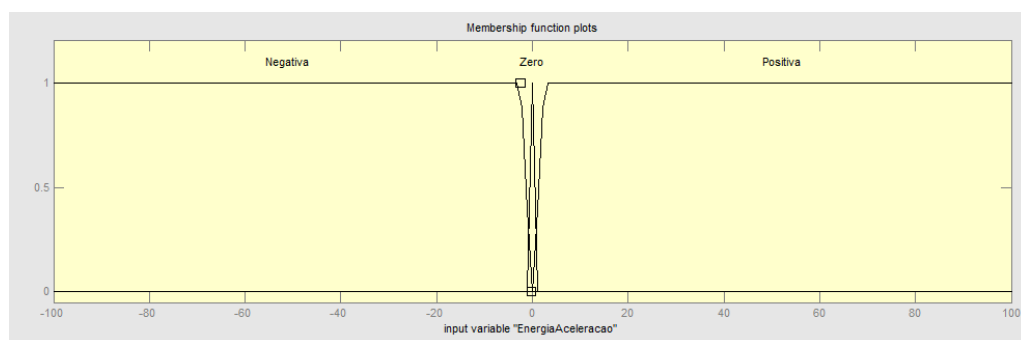


Figura 4.3: Função de pertinência do bloco de lógica difusa para Energia de Aceleração

Estas funções de pertinência foram definidas após ensaios para se determinar de que modo o tempo parado mudava nos diferentes ciclos de condução, bem como para verificar de que forma a energia da aceleração se comportava para variações de velocidade.

De seguida, apresentam-se as funções de pertinência da saída:

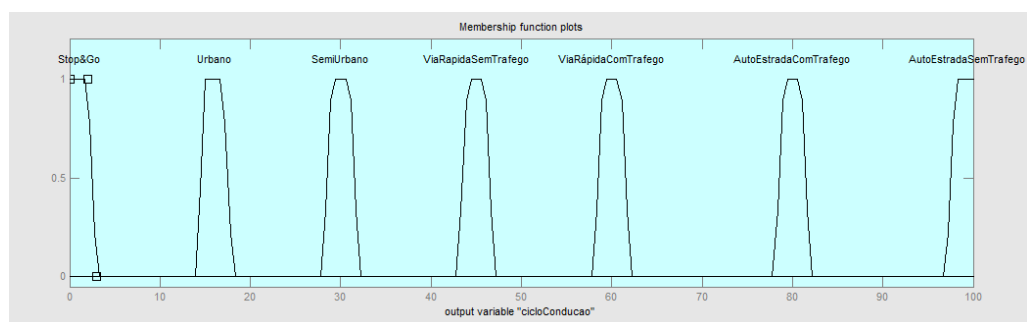


Figura 4.4: Função de pertinência do bloco de lógica difusa para a saída

Estas funções de pertinência foram ajustadas de forma a que os ciclos mais aproximados ficassem idênticos. Assim, o valor da saída aumenta com o aumento da velocidade média, sendo que os ciclos com elevado tráfego são também colocados consecutivamente, visto apresentarem propriedades semelhantes.

Considerando estas funções de pertença, apresentam-se, agora, as regras usadas para inferir o tipo de ciclo de condução. Estas regras foram construídas tendo em consideração as relações empíricas referidas anteriormente. Sublinhe-se que na tabela 4.2 o peso das regras para vMedium é de $\frac{1}{2}$.

Tabela 4.1: Regras usando velocidade média e aceleração

		Velocidade Média	
		vHigh	vVeryHigh
Energia de aceleração	Positiva	Via Rápida Com tráfego	Auto-Estrada Com tráfego
	Zero	Via Rápida Sem tráfego	Auto-Estrada Sem tráfego
	Negativa	Via Rápida Com tráfego	Auto-Estrada Com tráfego

Tabela 4.2: Regras usando velocidade média e tempo parado

		Velocidade Média	
		vLow	vMedium
Tempo parado	Baixo	Urbano	Semi Urbano
	Alto	Stop and Go	Semi Urbano

Como resultado deste processo obteve-se um valor, o qual, quando comparado com a seguinte tabela, nos permite identificar o tipo de ciclo.

Tabela 4.3: Escalas para identificação de ciclo de condução

Ciclo	Intervalo
Stop and Go	[0; 10[
Urbano	[10; 25[
Semi Urbano	[25; 40[
Via Rápida Sem tráfego	[40; 60[
Via Rápida Com tráfego	[60; 75[
Auto-Estrada Sem tráfego	[75; 90[
Auto-Estrada Com tráfego	[90; 100]

Por fim, o diagrama do algoritmo desenvolvido pode ser encontrado em [A.1](#).

4.1.3 Testes do Algoritmo

Para validar o algoritmo encontrado fizeram-se alguns testes com alguns dos ciclos já utilizados. Abaixo, apresentam-se as figuras nas quais é possível visualizar a identificação para diferentes ciclos.

- CYC MANHATTAN

Neste primeiro ciclo é possível constatar que se trata, principalmente, de um ciclo *Stop and Go*, e que, quando a velocidade é superior a 5km/h , durante um período de tempo mais alargado, tem-se um ciclo Urbano.

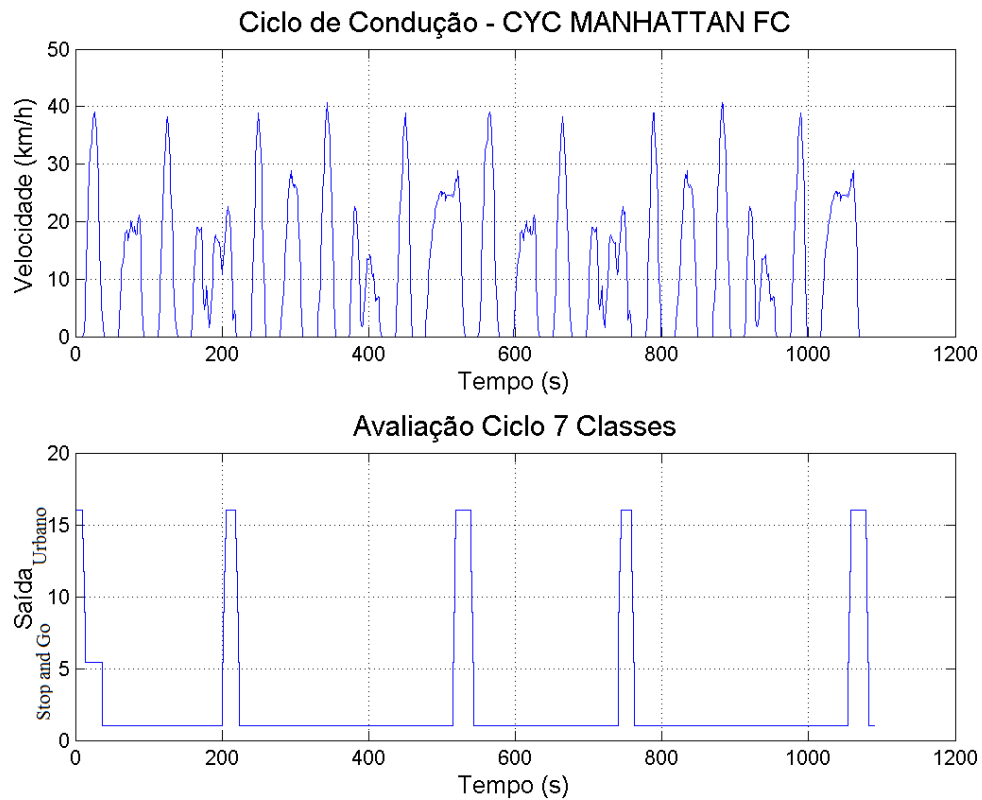


Figura 4.5: Teste de identificação de ciclo para CYC MANHATTAN

- CYC UDDS

Neste segundo ciclo é possível observar um comportamento semelhante ao anterior, contudo, existe, ainda, uma zona em que se assinala via rápida sem tráfego.

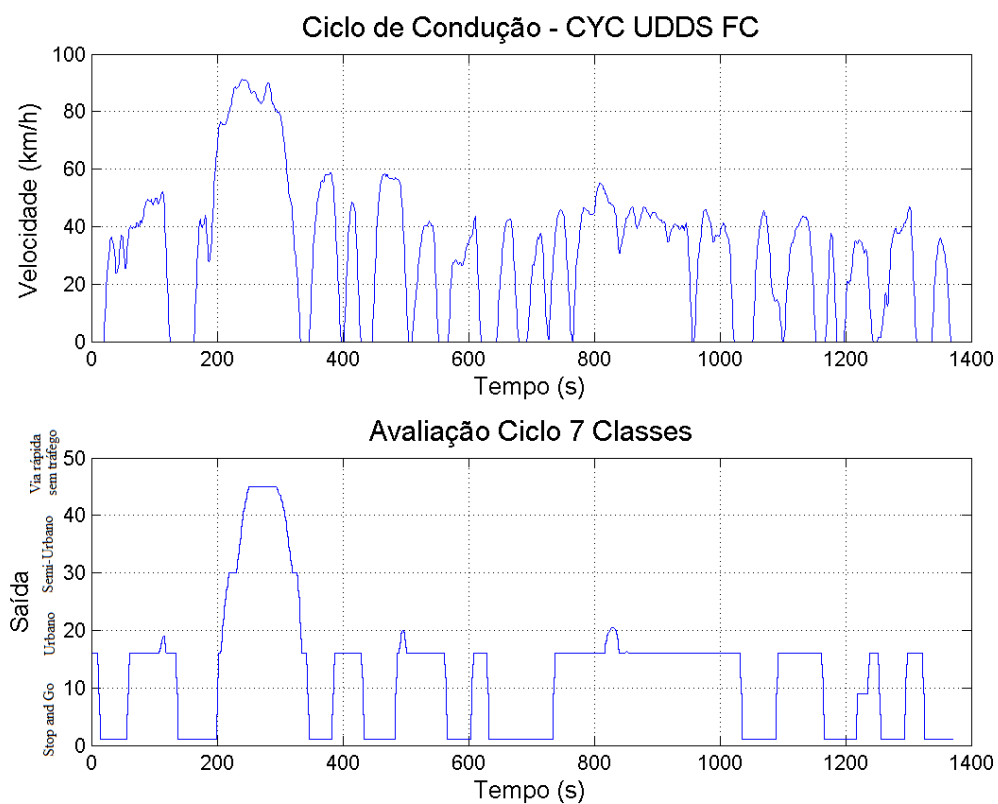


Figura 4.6: Teste de identificação de ciclo para CYC UDDS

- CYC HWFET

Neste terceiro ciclo é possível identificar zonas de ciclo Semi-Urbano e zonas de Via rápida sem tráfego.

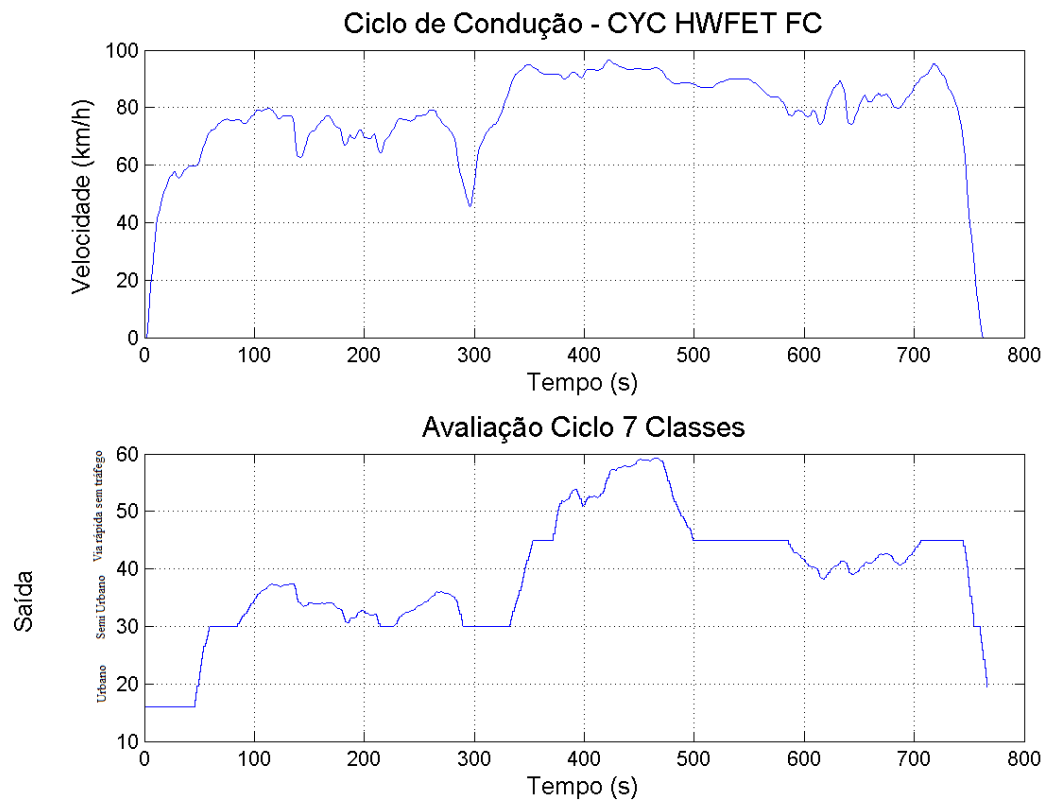


Figura 4.7: Teste de identificação de ciclo para CYC HWFET

- CYC CLEVELAND 90

Neste quarto ciclo é possível identificar zonas de ciclo Semi Urbano e zonas de Via rápida com tráfego.

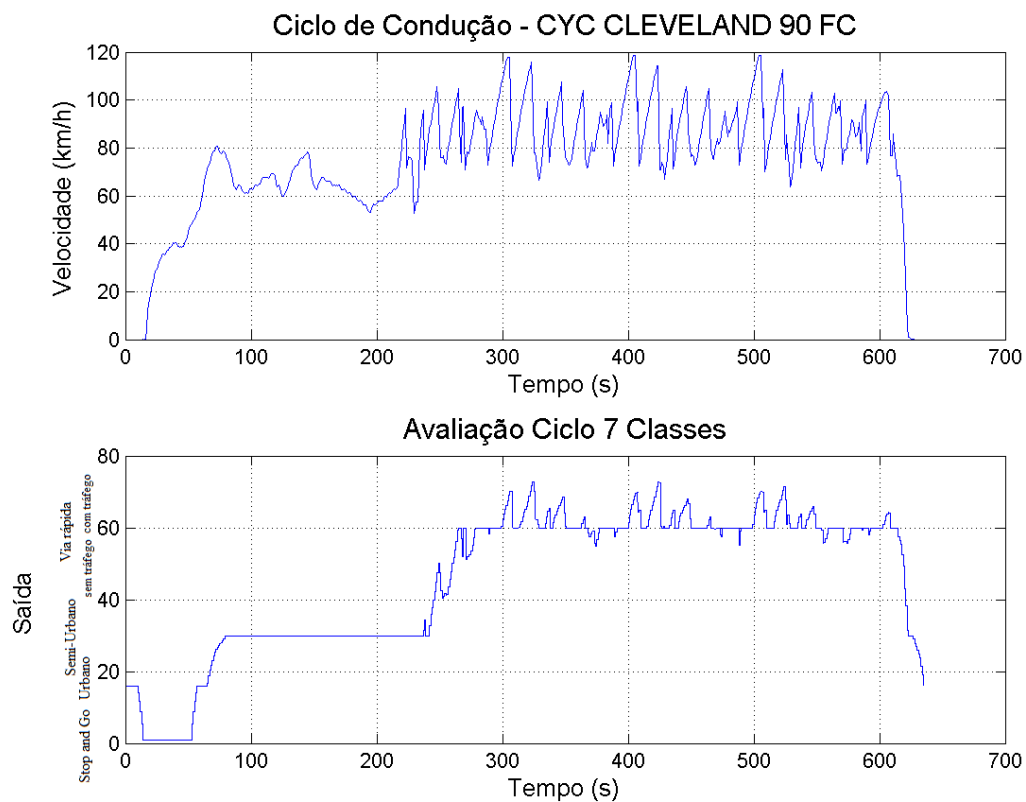


Figura 4.8: Teste de identificação de ciclo para CYC CLEVELAND 90

- CYC CLEVELAND

Neste quinto ciclo é possível identificar zonas de ciclo SemiUrbano e zonas de Auto Estrada com tráfego.

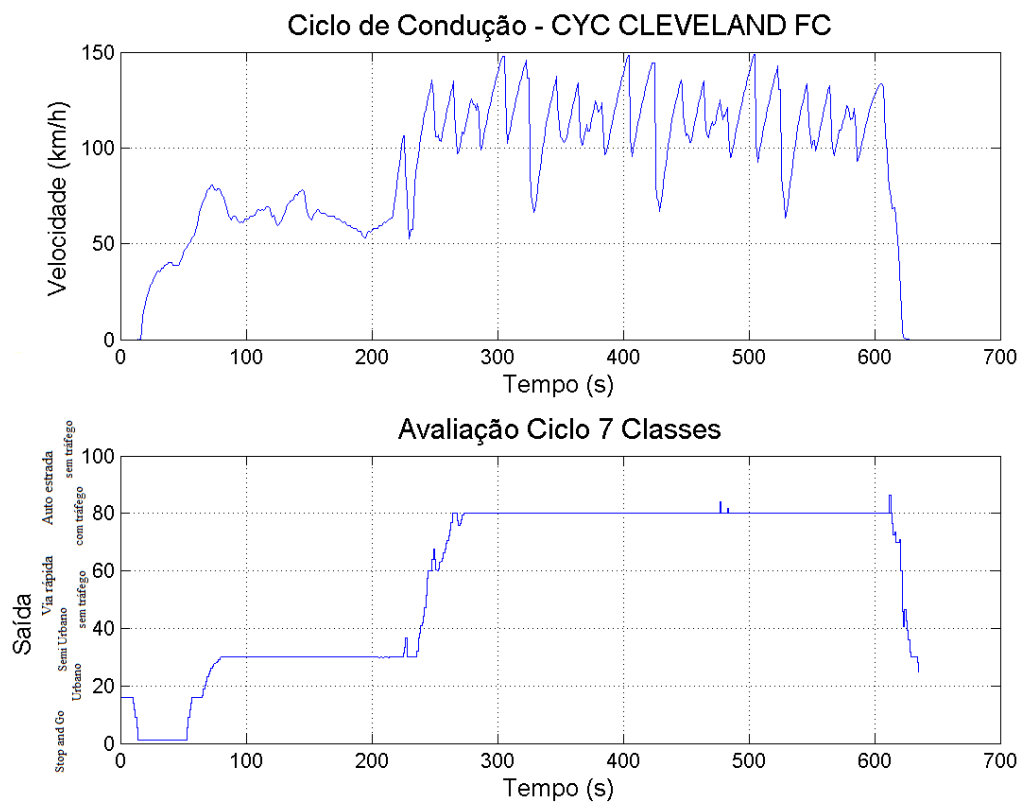


Figura 4.9: Teste de identificação de ciclo para CYC CLEVELAND

- CYC HWFET 110

Neste sexto ciclo o principal objetivo é o de mostrar a diferença entre Auto Estrada com tráfego e Auto Estrada sem tráfego.

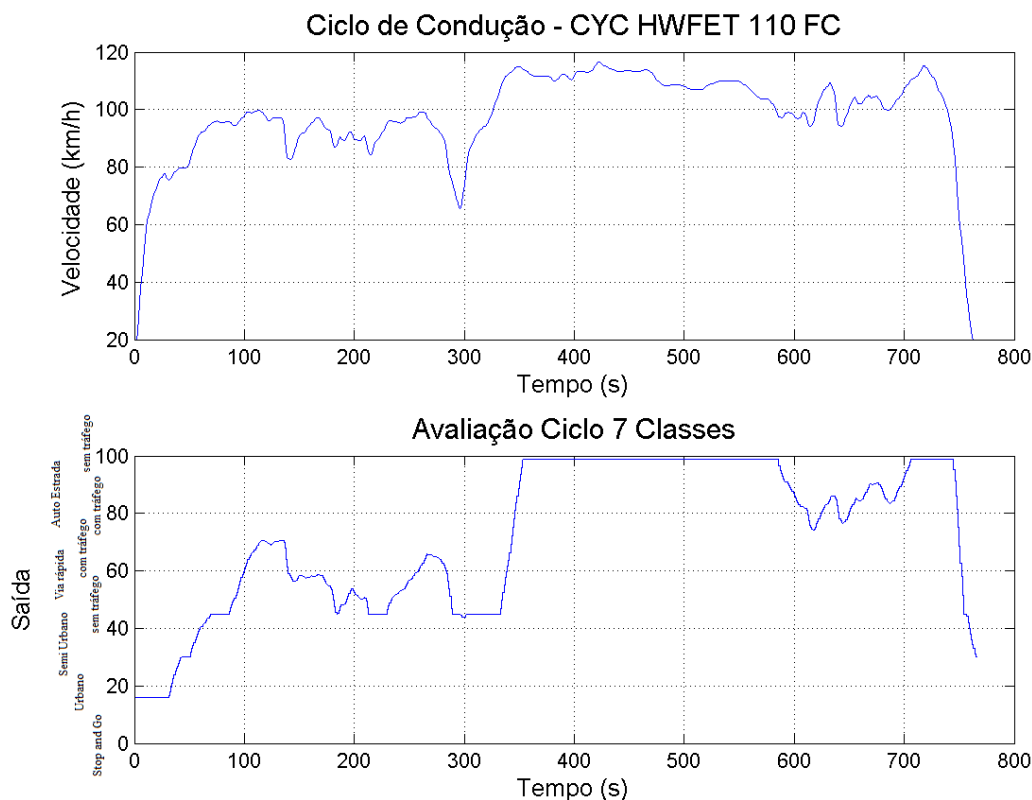


Figura 4.10: Teste de identificação de ciclo para CYC HWFET 110

4.2 Algoritmo de Estimativa de Consumo

Visa-se, nesta secção, apresentar o algoritmo desenvolvido para estimativa do consumo de combustível. Este terá por base as observações realizadas em 3.4 e 4.1.

4.2.1 Equações Matemáticas do Estimador

Considerando a análise já realizada, pretende-se, agora, afetar a eficiência do modelo de acordo com o tipo de Ciclo de condução. Irá, também, ser considerado que $\theta = 0$ em todas as ocasiões, obtendo-se, assim, as novas equações do modelo.

$$F_w = (m + m_j)a + \frac{1}{2}C_d A \rho v^2 + mgC_r + mg \quad (4.3)$$

$$ft = ft_{idle} + \frac{\delta}{\eta H_g} (mgC_r + (m + m_j)av + \frac{1}{2}C_d A \rho v^3) \quad (4.4)$$

$$\eta = f(CC) \quad (4.5)$$

$$\delta = \begin{cases} 1 & F_w > 0 \\ 0 & F_w \leq 0 \end{cases} \quad (4.6)$$

Tabela 4.4: Saídas do modelo final

Saída	Descrição	Unidades
F_w	Resistência total	N
ft	Consumo instantâneo	cm^3/s
δ	Controlo de consumo quando veículo parado	-
η	Eficiência variável	-

Tabela 4.5: Entradas do modelo final

Entrada	Descrição	Unidades	Obtenção	Opção de obtenção
a	Aceleração do veículo	m/s^2	Acelerómetro	GPS
v	Velocidade do veículo	m/s	Rede CAN do veículo	GPS/Acelerómetro
CC	Tipo de ciclo de condução	-	Estimativa	-

Tabela 4.6: Parâmetros do modelo 2

Parâmetro	Descrição	Unidades
g	Aceleração gravítica	m/s^2
m	Massa do veículo	kg
m_j	Massa equivalente de a inércia e das partes em movimento	kg
H_g	Calor equivalente do combustível	J/cm^3
ft_{idle}	Taxa de consumo em ponto morto	cm^3/s
ρ	Densidade do ar	kg/m^3
A	Área frontal do veículo	m^2
C_d	Coefficiente de arrasto de ar	-
C_r	Coefficiente de perda de rolamento	-

A principal diferença neste modelo, relativamente ao modelo 2, é a introdução da informação do ciclo de condução no cálculo de η . De facto, η deixa de ser considerado um parâmetro, passando a ser uma saída do modelo. Esta eficiência irá ser uma entrada no cálculo do consumo instantâneo. Apesar de não estar relatado nas equações referenciadas, foi ainda implementado um limite máximo para o consumo, que, no caso particular deste veículo, tomou o valor de $0.007L/s$.

A função de η é, então, definida em 4.2.1, onde CC é o ciclo de condução obtido de acordo com o apresentado em 4.1 e η_i é a eficiência obtida através da calibração já realizada, ou seja, 0.24.

$$\eta = \begin{cases} \eta_i + \frac{CC}{1000} & CC = \text{Stop and Go} \\ \eta_i + \frac{CC}{2000} & CC = \text{Urbano} \\ \eta_i + \frac{CC}{6000} & CC = \text{Semi-Urbano} \\ \eta_i - \frac{CC}{3000} & CC = \text{Via rápida sem tráfego} \\ \eta_i + \frac{CC}{8000} & CC = \text{Via rápida com tráfego} \\ \eta_i + \frac{CC}{1500} & CC = \text{Auto estrada com tráfego} \\ \eta_i + \frac{CC}{6000} & CC = \text{Auto estrada sem tráfego} \end{cases} \quad (4.7)$$

Esta função foi obtida recorrendo, novamente, à plataforma MATLAB/Simulink, para uma calibração semelhante à realizada em 3.2.2, sendo o principal objetivo melhorar a estimativa em todos os troços de ciclos de condução. Esta calibração foi executada recorrendo a pequenos ciclos de condução já apresentados, sendo que no final se quis verificar os resultados para ciclos de condução mais complexos. O diagrama Simulink do modelo desenvolvido pode ser encontrado em A.2.

4.2.2 Resultados do Estimador

Nesta sub secção serão apresentados os resultados da utilização do novo algoritmo, tendo por base os resultados obtidos na calibração anterior.

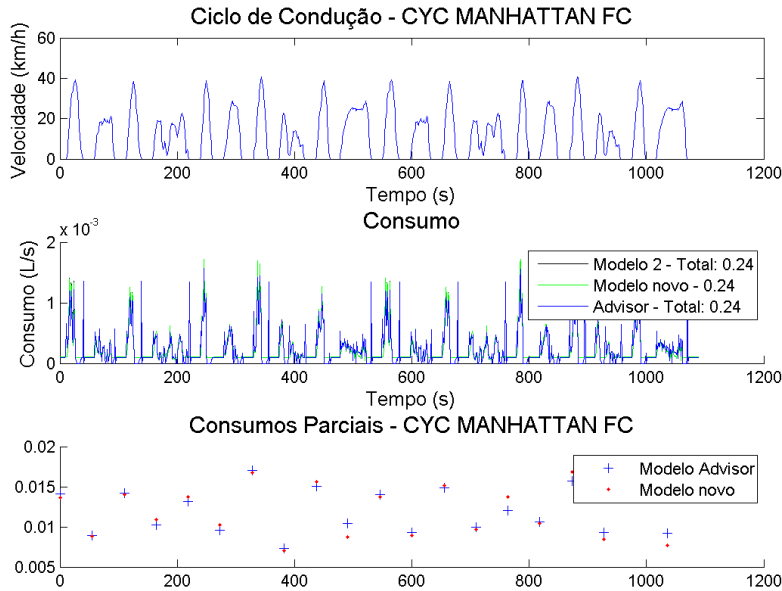


Figura 4.11: Resultados do novo algoritmo para o ciclo MANHATTAN

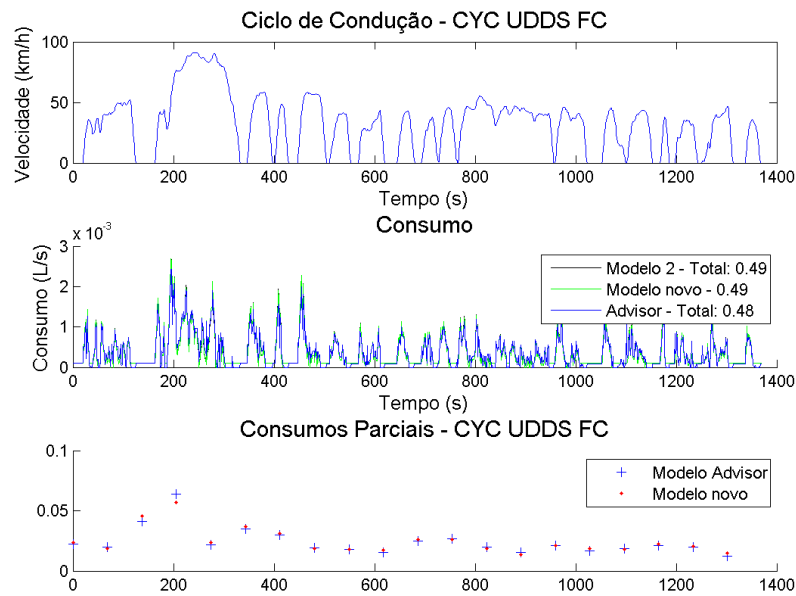


Figura 4.12: Resultados do novo algoritmo para o ciclo UDDS

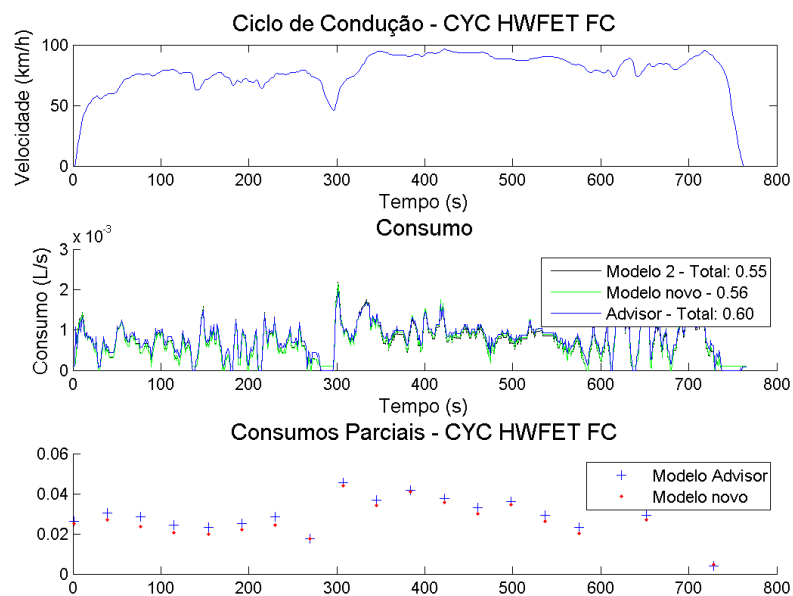


Figura 4.13: Resultados do novo algoritmo para o ciclo HWFET

Com base nas três últimas figuras, é possível verificar que as melhorias introduzidas pelo novo modelo não são significativas para estes ciclos, contudo é possível constatar que existem pequenas

melhorias, o que será indicativo que, para ciclos diferentes dos usados na calibração se poderá obter resultados mais satisfatórios.

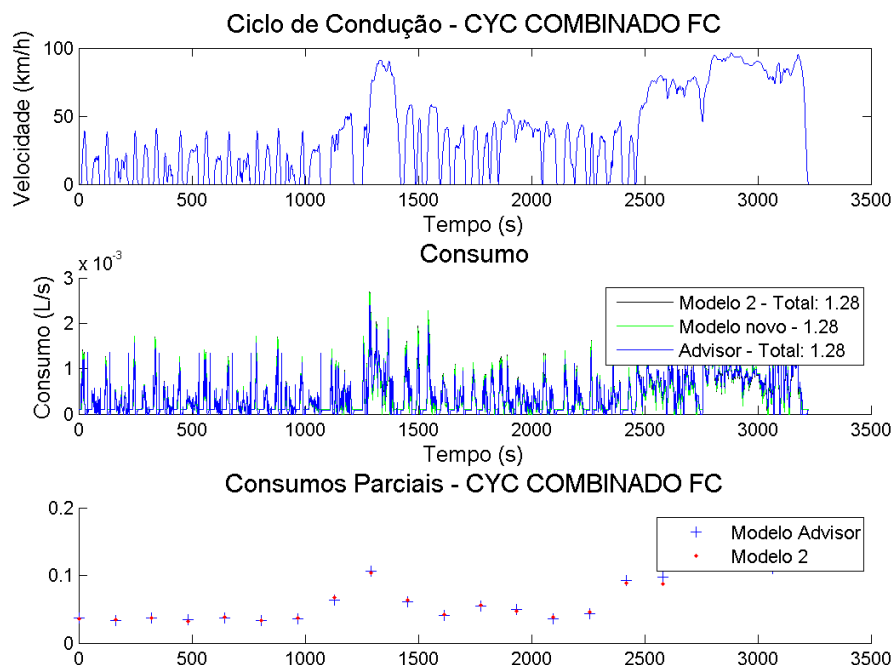


Figura 4.14: Resultados do novo algoritmo para o ciclo COMBINADO

A figura anterior mostra que combinando os três ciclos já apresentados (MANHATTAN, UDDS, HWFET) se obtêm resultados semelhantes para os dois modelos, podendo tirar-se conclusões semelhantes às anteriores.

Na figura 4.15 é possível verificar que, ao contrário das precedentes, existe uma grande melhoria com a introdução deste novo modelo. Este resultado torna óbvio que este irá ter mais impacto a grandes velocidades e nas quais as acelerações são mais sentidas.

No gráfico 4.16 é possível visualizar que, para este ciclo, obtido recorrendo ao *smartphone*, é possível verificar que, caso este seja simulado pelo Advisor, os resultados para o nosso modelo são bastante positivos, tal como nos casos dos ciclos já presentes no Advisor.

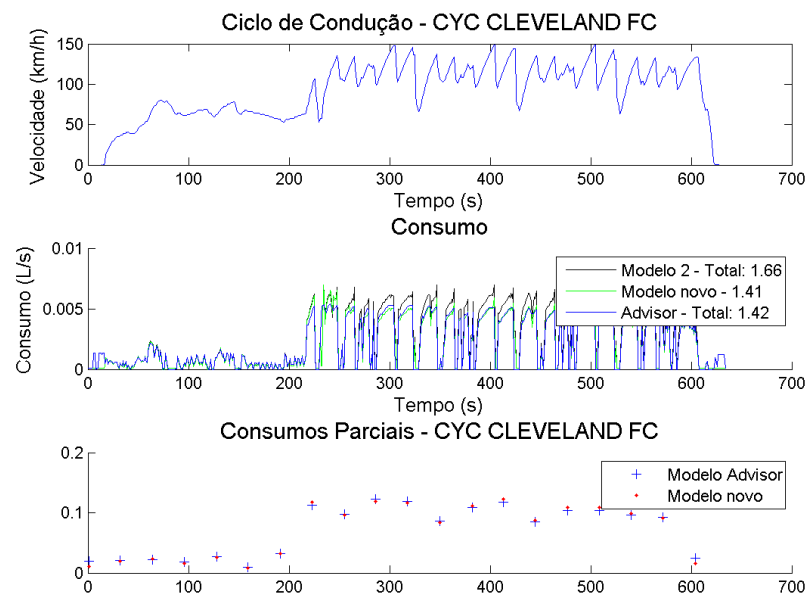


Figura 4.15: Resultados do novo algoritmo para o ciclo CLEVELAND

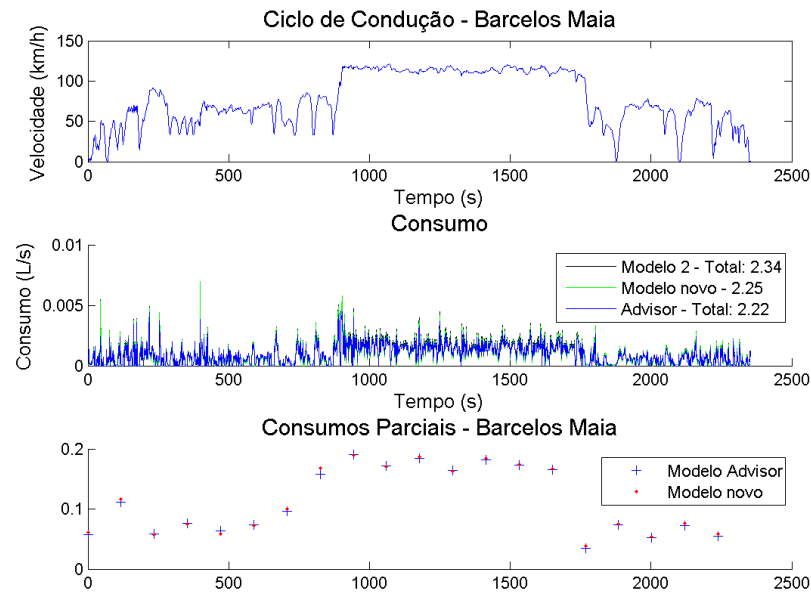


Figura 4.16: Resultados do novo algoritmo para o ciclo Barcelos Maia

4.3 Conclusões

O recurso à utilização de um sistema de lógica difusa, cujas entradas são a velocidade média, o tempo parado e a energia da aceleração média, permite a conceção de um algoritmo capaz de categorizar o tipo de ciclo em sete classes, de forma bastante fiável. A sua saída foi compreendida no intervalo $[0;100]$ e foi dividida em escalões para a identificação. Esta foi usada para variar a eficiência do estimador de consumo de combustível, tendo, ainda, em linha de conta o tipo de ciclo que se encontrava a percorrer. Seguidamente, simulou-se este modelo, obtendo-se uma nova estimativa de consumo e concluiu-se que, apesar de poder introduzir pequenas melhorias em situações específicas, a evolução do algoritmo inicial permite, principalmente, melhorar a *performance* do estimador em situações de maior *stress*. O passo seguinte será o de avaliar se em casos reais (veículos reais) o mesmo efeito será sentido.

Capítulo 5

Implementação e Teste do Algoritmo em Sistema Operativo Android

Considerando o objetivo último deste projeto, que é o de desenvolver um algoritmo capaz de estimar o consumo instantâneo de combustível de um automóvel num dispositivo móvel, implementou-se o algoritmo desenvolvido em 4 num dispositivo Android, aplicação esta que irá ser apresentada neste capítulo.

Assim, discutir-se-ão as variáveis a adquirir e o método utilizado, bem como o algoritmo implementado para a estimativa do consumo.

5.1 Aquisição de Entradas

Nesta secção inscrevem-se informações relevantes, relativas ao método de aquisição de dados, bem como ao respetivo armazenamento. Assim sendo, tendo por base os aparelhos referenciados em 2.3, pretende-se realizar uma análise e desenvolver um método eficiente de guardar dados obtidos através dos sensores do *smartphone*. Estes serão utilizados para que, posteriormente, se possa analisar o desempenho do estimador de consumo, bem como, numa fase mais adiantada do projeto, alertar o condutor para a sua condução.

5.1.1 Variáveis a adquirir

Para uma estimação correta do consumo instantâneo do combustível, com base no modelo desenvolvido em 4.2, é necessário conhecer a velocidade do veículo e a sua aceleração.

Tendo em vista um dos objetivos deste trabalho, serão apenas usados os sensores disponíveis num *smartphone*, nomeadamente o recetor GPS. Com base nestes sensores, é possível conhecer a velocidade e a aceleração. Por outro lado, recorrendo ao GPS é possível obter diretamente a velocidade e derivando esta, de acordo com as regras já estabelecidas, obtém-se a aceleração. Numa fase posterior, poder-se-ia considerar a utilização do acelerómetro para uma obtenção direta da aceleração. Contudo, seria necessário realizar um estudo aprofundado de todas as acelerações envolvidas no deslocamento do automóvel e no ruído que este sensor poderia produzir.

Para uma análise subsequente, serão guardados o instante de amostragem, a posição global do veículo com base no GPS, a velocidade, a aceleração e a altitude, sendo que a variação da última pode, através de manipulação, apresentar-nos a inclinação.

5.1.1.1 Frequências de amostragem

Utilizando o modelo desenvolvido, sabe-se que a estimativa será realizada a cada segundo. Deste modo, idealmente, todos os sinais deveriam ser amostrados com uma frequência de 1Hz, uma vez que, com a diminuição da frequência e o aumento do período de amostragem, a qualidade na estimativa do consumo piora.

O GPS comum, segundo [33] terá uma precisão de 10 m, pois apenas temos disponível o serviço *Standard Positioning Service*, requerendo a existência de quatro satélites visíveis. A frequência mínima de amostragem é dependente do dispositivo. No sistema operativo Android está definido que, para obtenção de resultados de GPS, se deve estabelecer o período mínimo de amostragem, tal como a distância mínima percorrida. Donde, caso estes valores tomem o valor de 0, o dispositivo Android disponibilizará a nova informação sempre que possível. Para provocar uma aquisição de frequência fixa, utilizaram-se *timers* para verificação do estado da informação do GPS a cada segundo, ignorando, desta maneira, a informação que existisse caso este tivesse várias alterações durante um segundo. O conceito usado é semelhante ao de um *Sample and Hold*.

5.1.2 Método de gravação de dados

Considerando as variáveis necessárias, isto é, a aceleração, a altitude, a longitude e a latitude, é preciso obtê-las, para cada instante de tempo. Como primeira hipótese, ponderou-se o armazenamento usando codificação XML, contudo, após análise de [25], percebe-se que o tamanho do ficheiro aumenta cerca de 180 vezes. Assim, optou-se por guardar os dados num ficheiro de texto, em que no topo serão apresentados os nomes das variáveis, separados por um TAB. Os valores serão depois apresentados ao longo das linhas, também eles separados por um TAB.

O ficheiro será guardado na memória do *smartphone* para que possa, posteriormente, ser analisada a estimativa e seja possível avaliar, tanto o desempenho do condutor, como o do estimador.

Na figura seguinte pode-se observar um ficheiro típico

2014-4-9_13-21-21 - Bloco de notas

Ficheiro

Editar

Formatar

Ver

Ajuda

Time	Longitude	Latitude	Altitude	Speed	Acceleration		
13:21:22	-8.	41.	3	175.80	16.250	-0.313	
13:21:23	-8.	0	41.	4	175.80	16.188	-0.063
13:21:24	-8.	6	41.	0	175.20	16.250	0.063
13:21:25	-8.	1	41.	0	174.50	15.688	-0.563
13:21:26	-8.	6	41.	2	173.60	15.688	0.000
13:21:27	-8.	8	41.	4	173.00	16.250	0.563
13:21:28	-8.	3	41.	0	172.80	16.188	-0.063
13:21:29	-8.	3	41.	0	172.40	15.563	-0.625
13:21:30	-8.	5	41.	0	172.10	14.500	-1.063
13:21:31	-8.	9	41.	7	172.10	13.688	-0.813

Figura 5.1: Ficheiro exemplificativo do sistema de aquisição

Os ficheiros serão armazenados na memória interna do *smartphone* e terão a denominação de acordo com a data e hora em que a aquisição se inicia. Assim, os ficheiros terão o seguinte formato: AAAA-MM-DD_HH-MM-SS.txt, um exemplo do nome do ficheiro pode ser encontrado na figura 5.1.

5.1.3 Tamanhos exemplificativos dos ficheiros

Na seguinte tabela é possível observar de que forma o tamanho dos ficheiros a armazenar varia com o aumento do tempo. O mesmo poderá ser observado graficamente através da figura 5.2.

Tabela 5.1: Tamanhos exemplificativos dos ficheiros

Tempo	Bytes	Kilobytes	MegaBytes
1 minuto	2861	2.79	0.003
5 minutos	14305	13.97	0.014
10 minutos	28610	27.94	0.027
1 hora	171660	167.64	0.164
2 horas	343320	335.27	0.327
5 horas	858300	838.18	0.818

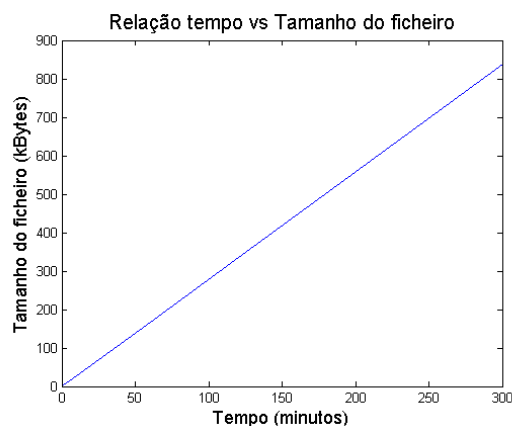


Figura 5.2: Relação entre tamanho do ficheiro e tempo de aquisição

5.2 Interface gráfica

A interface, para o utilizador da aplicação, apresentará, em grande plano, o seu consumo instantâneo e a sua velocidade. Em segundo plano, será evidenciada a posição global, o estado atual, o consumo total e a distância percorrida. Por fim, existirá um botão que irá permitir iniciar e parar a aquisição. Atente-se que o ciclo de condução será escondido, não sendo este, na aplicação final, apresentado diretamente.

Na figura seguinte é possível observar a interface construída.



Figura 5.3: Interface Android

5.3 Algoritmo implementado

O algoritmo implementado em Android, presente em [B](#), pretende ser uma réplica fiel do simulado em MATLAB/Simulink. Assim, dividiu-se o algoritmo em várias fases. A fase de aquisição, na qual se aplica o método já referido em [5.1.1.1](#). Com base na aquisição realizada e como explicitado anteriormente, verifica-se o tipo de ciclo de condução, com o intuito de variar a eficiência do automóvel. Finalmente, recorre-se às funcionalidade do Java e calcula-se o consumo, apresentando-o na interface. Os valores são guardados no ficheiro, como indicado anteriormente.

Na fase de aquisição, após testes realizados, verificou-se que, por vezes, existem grandes variações no valor da velocidade. Estas têm uma dinâmica muito rápida, pelo que se conclui que não podem ocorrer num veículo. O fenómeno aqui referido será apresentado em [5.3.1](#). Para o combater introduziu-se um filtro no qual se impõe um limite para a variação máxima, a saber:

$$v[t] = \begin{cases} v[t-1] & v[t] - v[t-1] > 2 \\ v[t-1] & v[t] - v[t-1] < -4 \\ v[t] & -4 < v[t] - v[t-1] < 2 \end{cases} \quad (5.1)$$

Na fase de determinação do ciclo recorreu-se a uma biblioteca de Fuzzy para implementação do sistema de lógica difusa em Java, esta pode ser encontrada em [34, 35].

Todas as outras fases são atingidas recorrendo às bibliotecas do Java e do Android.

De destacar que a aquisição só pode ser iniciada após existir sinal de GPS, caso contrário o sistema não irá permitir o início da aquisição. Por fim, o algoritmo foi implementado e testado num dispositivo Samsung Galaxy S3.

5.3.1 Resultados da aquisição

Nesta secção serão apresentados os resultados do primeiro teste à aquisição da velocidade, usando um dispositivo Android. Serão, primeiramente, mencionados resultados para velocidades aproximadamente constantes e, de seguida, os resultados para percursos com velocidade variável. Todas as aquisições foram feitas com um período de amostragem (T_s) igual a 1.

Importa, ainda, referir que, após uma primeira análise dos dados obtidos, foi possível compreender que, por vezes, o sinal de GPS aparentava ter variações de velocidade superiores às possíveis na dinâmica do automóvel. Por este motivo, optou-se por filtrar estas variações para que sejam ignoradas.

Expõem-se, em primeiro lugar, os ciclos de condução conseguidos a velocidade constante. Em seguida será possível observar o comportamento da aquisição em percursos mais complexos e com velocidades variáveis, bem como perceber melhor a influência do filtro de velocidade apresentado em 5.3.

5.3.1.1 Aquisição a velocidade constante

- Velocidade 65 km/h

Nesta figura é possível constatar que, a maior parte do percurso, é feita, de acordo com o *smartphone*, a uma velocidade ligeiramente inferior de 65 km/h (velocidade presente no velocímetro), sendo aproximadamente 62/63 km/h. Registe-se, ainda, que, devido ao automóvel ter atravessado um túnel, aconteceu o fenómeno já referido em 5.3 relativo às variações de velocidade. Este foi atenuado pelo filtro implementado.

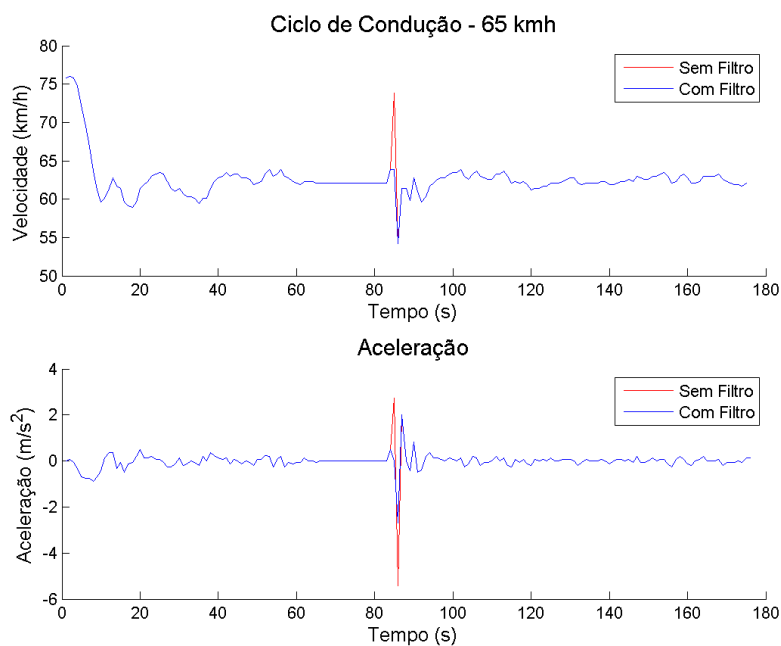


Figura 5.4: Ciclo obtido a 65 km/h

- Velocidade 70 km/h

Neste ciclo é possível ver-se que, após uma fase de aceleração, o veículo estabilizou perto da velocidade de 68 km/h, o que é ligeiramente inferior à velocidade apresentada no velocímetro (70 km/h). Todavia, o GPS do veículo em que o teste foi executado apresentava uma velocidade semelhante à apresentada pelo *smartphone*.

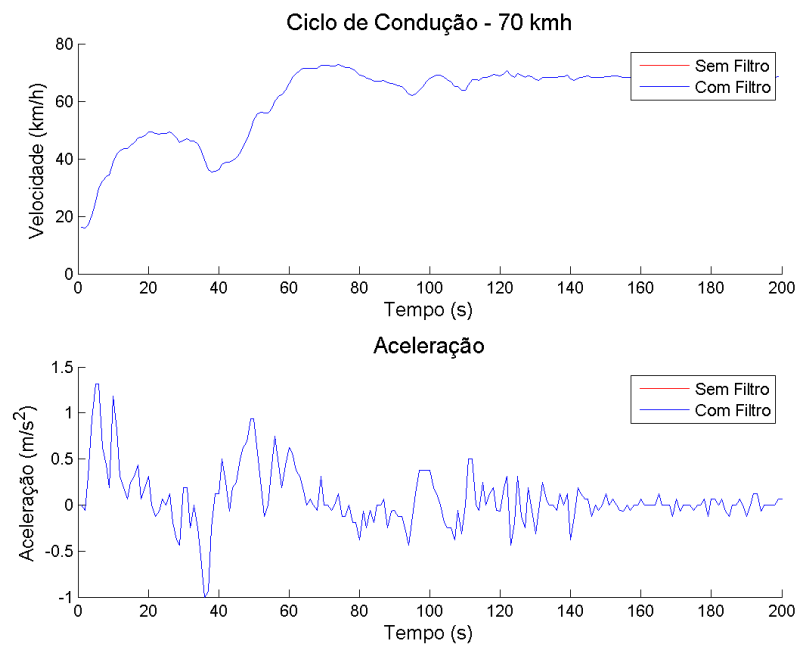


Figura 5.5: Ciclo obtido a 70 km/h

- Velocidade 75 km/h

Neste ciclo não existiu uma fase de estabilização durante a aquisição, optando-se por estabilizar a velocidade em 75km/h e, posteriormente, iniciar a aquisição. Os resultados estão de acordo com os anteriores, sendo possível observar que a velocidade foi, novamente, adquirida com um valor inferior a 75km/h.

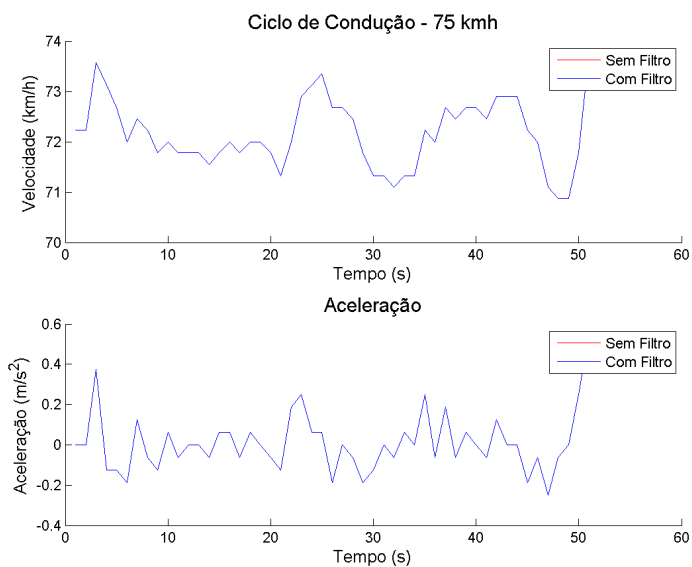


Figura 5.6: Ciclo obtido a 75 km/h

- Velocidade 100 km/h

A análise desta figura permite-nos perceber que se pode observar o mesmo efeito para a aquisição a 65 km/h.

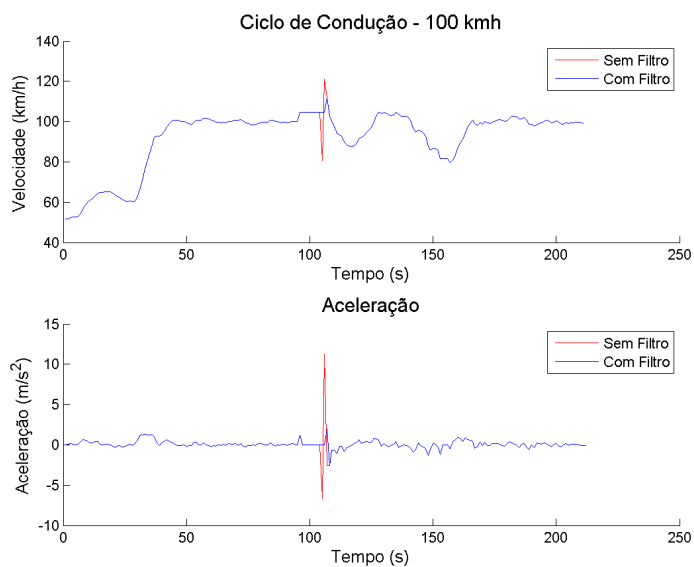


Figura 5.7: Ciclo obtido a 100 km/h

5.3.1.2 Aquisição a velocidade variáveis

- Viagem de Maia para FEUP

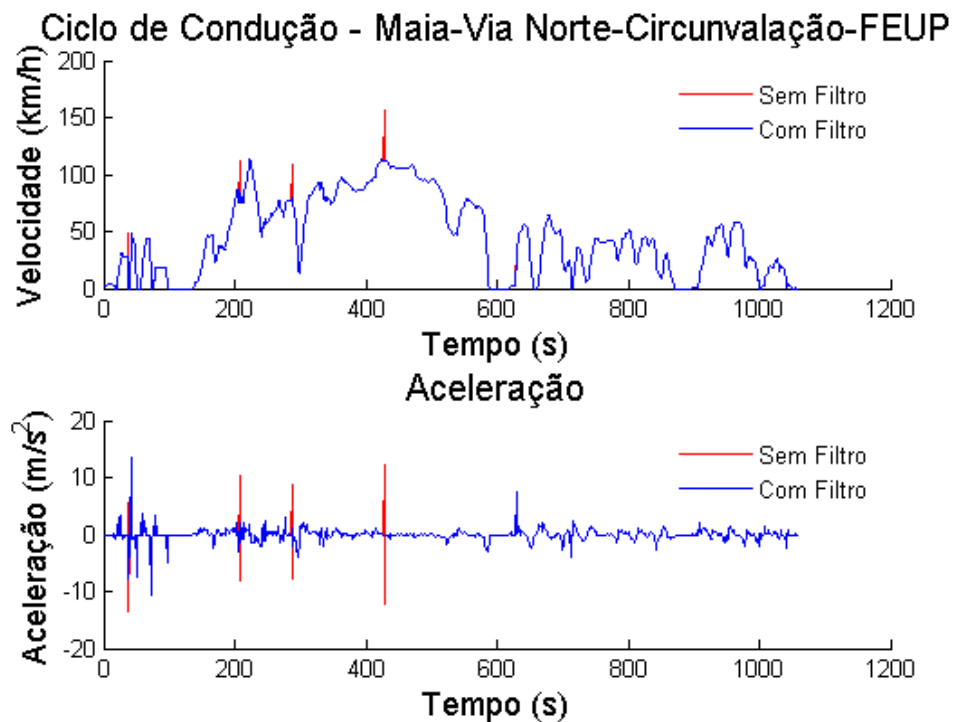


Figura 5.8: Ciclo entre Maia e FEUP

Neste ciclo de condução é possível verificar de que forma o filtro implementado ajuda na aquisição correta da velocidade. Para este ciclo traçou-se, ainda, o percurso no Google Maps, recorrendo às coordenadas extraídas durante o processo de aquisição. O resultado demonstra a correta aquisição das coordenadas. Refira-se que, por razões de privacidade, o início da viagem foi excluído.

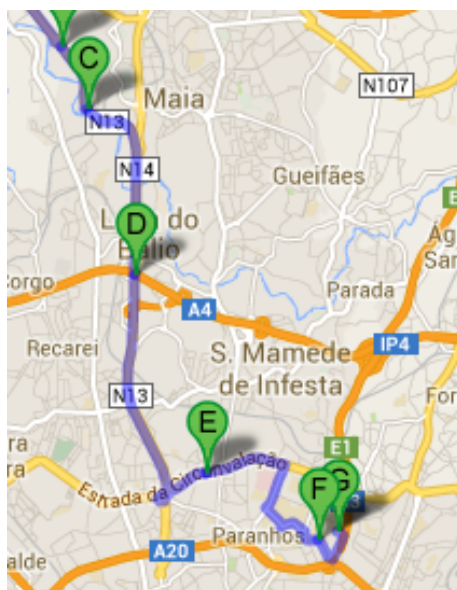


Figura 5.9: Ciclo entre Maia e FEUP - Localização GPS

- Viagem de FEUP para Maia

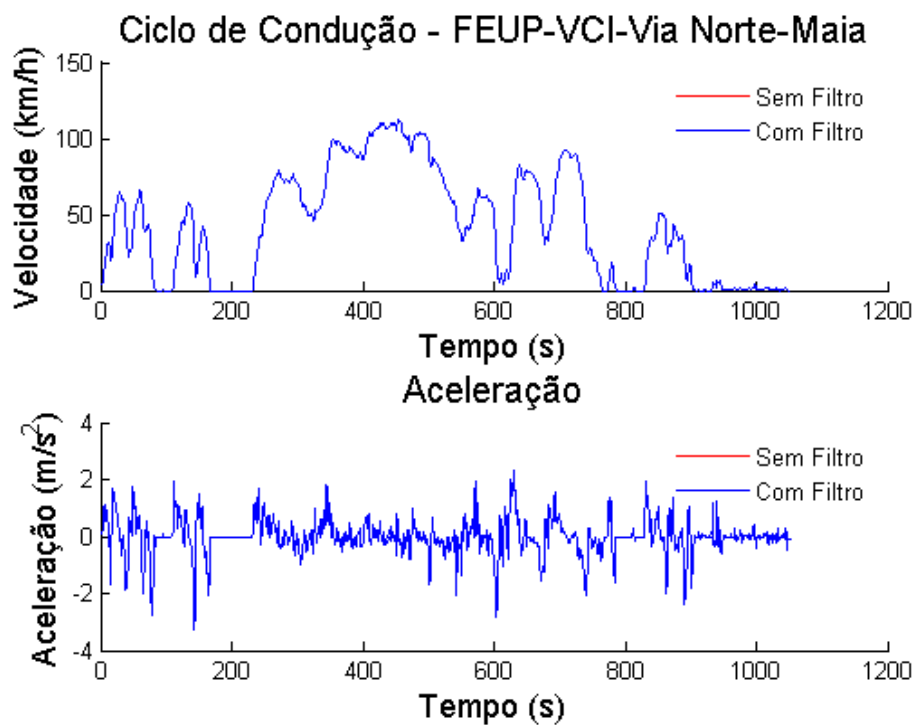


Figura 5.10: Ciclo entre FEUP e Maia

Esta figura mostra o percurso inverso ao anterior, sendo o objetivo demonstrar que funciona para vários tipos de ciclos e que o filtro nem sempre tem necessidade de atuar.

5.3.2 Resultados da identificação de ciclo

Aqui pretende-se mostrar os resultados obtidos para a identificação de ciclo, quando estes foram realizados no Android. Para tal, substitui-se a saída, "Altitude" dos dados, pelo valor do ciclo. Assim, será possível realizar gráficos, com recurso a MATLAB, para verificar de que forma se comporta a estimativa do ciclo. Igualmente, será possível comparar com os resultados obtidos pelo algoritmo de identificação simulado em MATLAB.

- Ciclo urbano

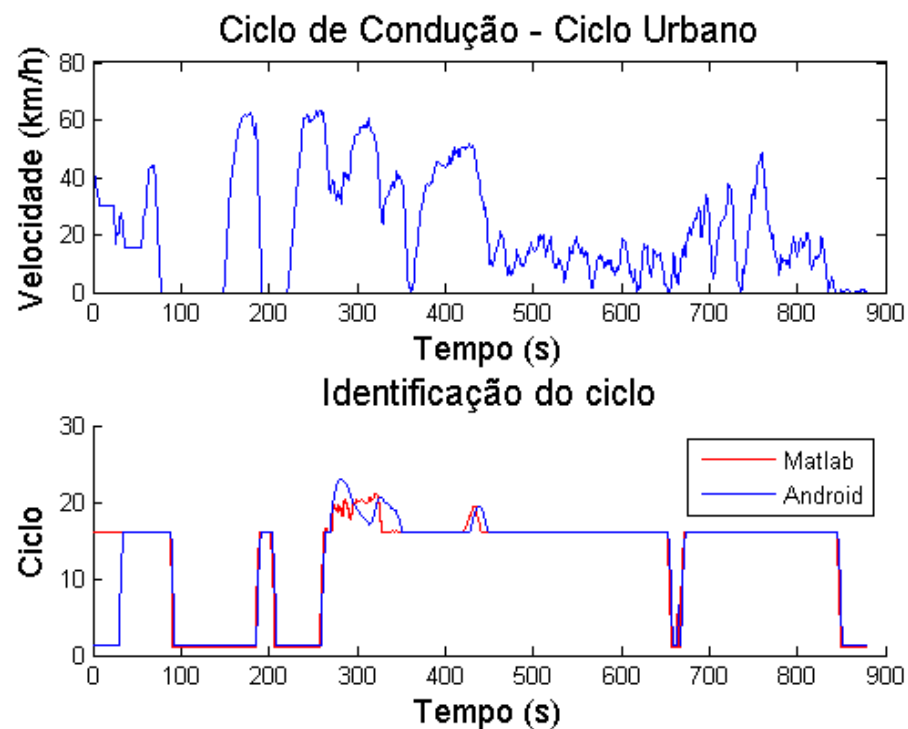


Figura 5.11: Ciclo Urbano

Neste ciclo é verifica-se que, para um ciclo urbano, a aplicação é capaz de distinguir entre Stop and Go e Urbano, tal como pretendido. É, também, possível observar que a identificação se encontra de acordo com a realizada pela simulação MATLAB/Simulink.

- Ciclo entre Foz e Moreira

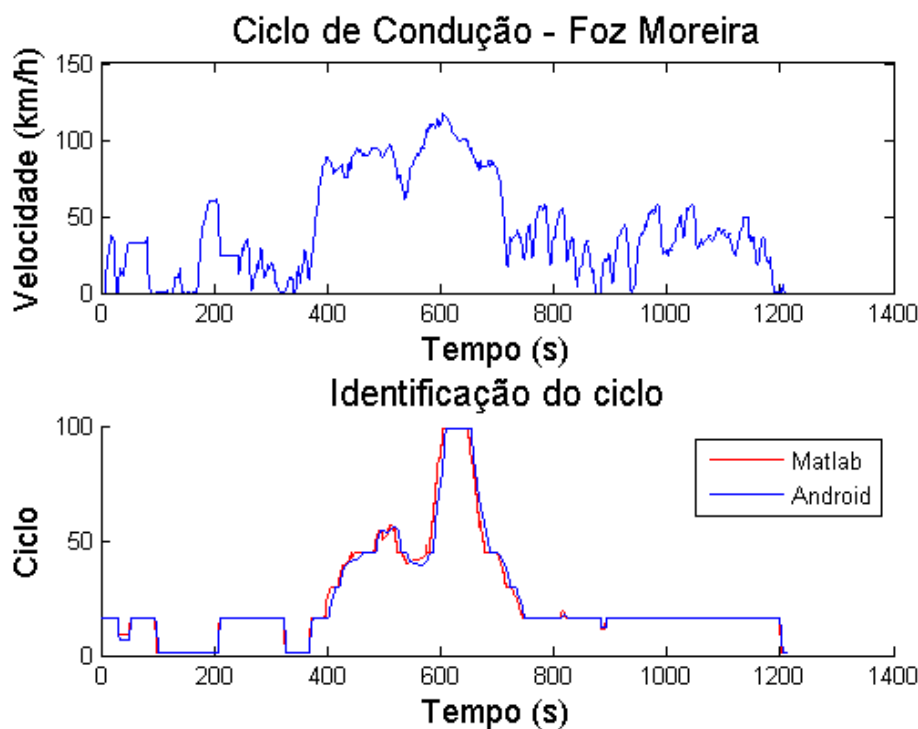


Figura 5.12: Ciclo entre Foz e Moreira

Neste ciclo pode proceder-se à identificação de vários tipos de ciclos diferentes, sendo possível, mais uma vez, verificar que o comportamento é bastante semelhante ao das simulações Matlab/Simulink. Observa-se, ainda, que em determinados momentos, o dispositivo perdeu o sinal GPS, pelo que, durante esses momentos, se manteve a última velocidade, pois esta é a melhor informação disponível.

5.3.3 Resultados da estimativa do consumo

Para o teste final, decidiu-se realizar um teste real, o qual envolvia a aquisição dos ciclos de condução, recorrendo à aplicação em Android e à comparação dos resultados da simulação com os resultados apresentados pelo visor de um veículo, que apresentasse já uma estimativa de consumo. Assim, usou-se um veículo Alfa Romeo MITO, que está dentro do tipo de veículo para o qual se pretende estimar o consumo e seguiram-se os seguintes passos:

1. Encher depósito;
2. Guardar todos os ciclos de condução realizados;
3. Anotar todos os consumos, por cada ciclo, disponibilizados pelo veículo;
4. Encher novamente o depósito e anotar a quantidade de combustível colocada;

5. Transferir os ciclos de condução para o computador e simular em MATLAB/Simulink;
6. Comparar os resultados de cada ciclo simulado com os ciclos obtidos no fim;
7. Comparar os resultados de consumos totais (Consumo real, consumo indicado pelo veículo, consumo indicado pelo estimador).

Considerando as etapas apresentadas e a necessidade de estimar o consumo do veículo, tornou-se imperioso parametrizar o veículo. Assim, recorrendo a [36] e ao manual do veículo, obtiveram-se os seguinte parâmetros:

Tabela 5.2: Parâmetros obtidos para o Alfa Romeo MITO

Parâmetro	Valor	Unidades
A	2.487	m^2
C_d	0.29	—
m	1215	kg
m_j	50	kg

A área frontal foi estimada equiparando a um retângulo cujo comprimento seria a largura do veículo e a largura corresponderia à altura do veículo. O coeficiente de arrasto foi retirado de [36] a m_j considerou-se igual ao anteriormente estipulado e a m foi tomada como a tara do veículo adicionada do peso típico do condutor.

Para além deste parâmetros, com base nas simulações, calibrou-se o modelo com os seguintes parâmetros:

Tabela 5.3: Parâmetros estimados para o Alfa Romeo MITO

Parâmetro	Valor	Unidades
η_i	0.29	m^2
$f_{t_{idle}}$	0.23	—

Todos os restantes parâmetros mantiveram-se constantes. De referir que para o modelo 2 $\eta_t = \eta_i$.

5.3.3.1 Apresentação de Resultados

Serão, agora, apresentados os resultados do teste final. Tome-se que foram colocados 15 litros de combustível no fim do teste e o consumo médio total foi de $6.5L/100km$, assim como a distância percorrida foi de $230.6km$, pelo que, segundo a estimativa do automóvel, gastaram-se: $14.99L$, o que está de acordo com o combustível colocado ($15L$).

Antes de mais, apresentam-se alguns dos trinta e cinco ciclos que compõe o ciclo total. A partir destes, como referido anteriormente, foi possível calibrar o modelo com os seus novos parâmetros, bem como verificar se os consumos eram aproximados para os vários ciclos.

- Ciclo 1

Neste primeiro ciclo verifica-se que a influência do novo modelo não é sentida de forma muito significativa, apresentando apenas uma pequena aproximação por parte do novo algoritmo. Esta comparação é realizada tendo em consideração que o consumo real médio foi de $5.5L/100km$.

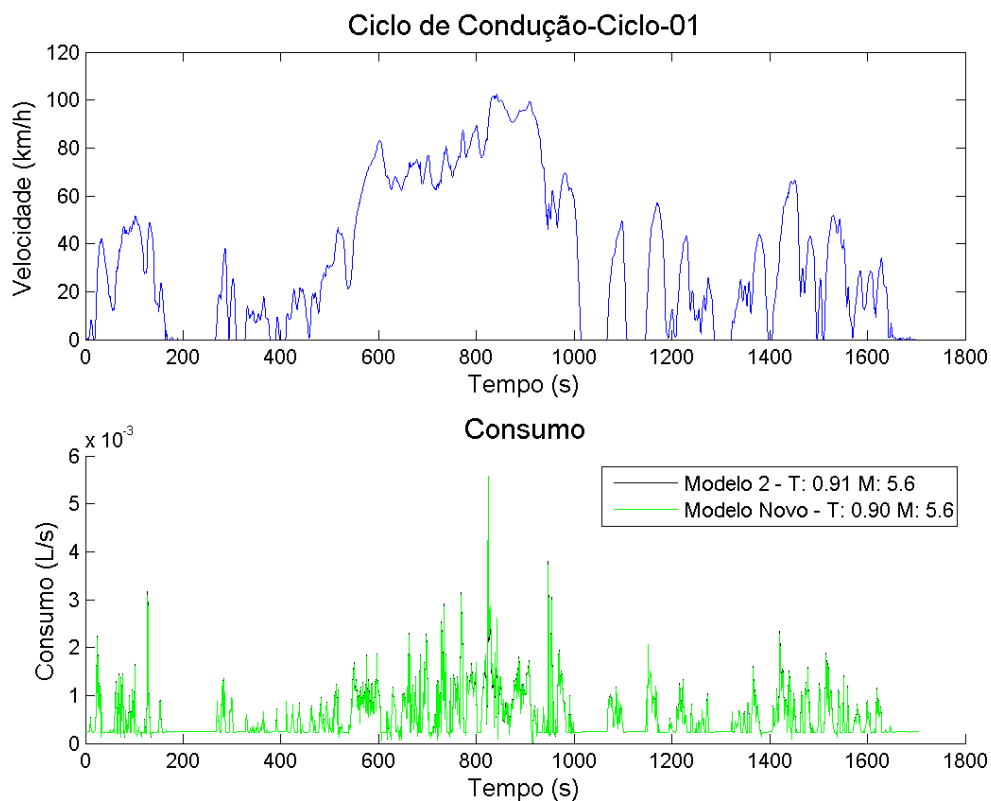


Figura 5.13: Apresentação do resultado final para ciclo 1

- Ciclo 10

Este décimo ciclo foi bastante útil para a calibração dos novos parâmetros do veículo, isto porque possui um grande intervalo de tempo em que a velocidade é nula. Assim, usou-se este ciclo como referência para calibrar o $f_{t_{idle}}$, sendo que, após a sua calibração, se verificava o comportamento para os restantes ciclos.

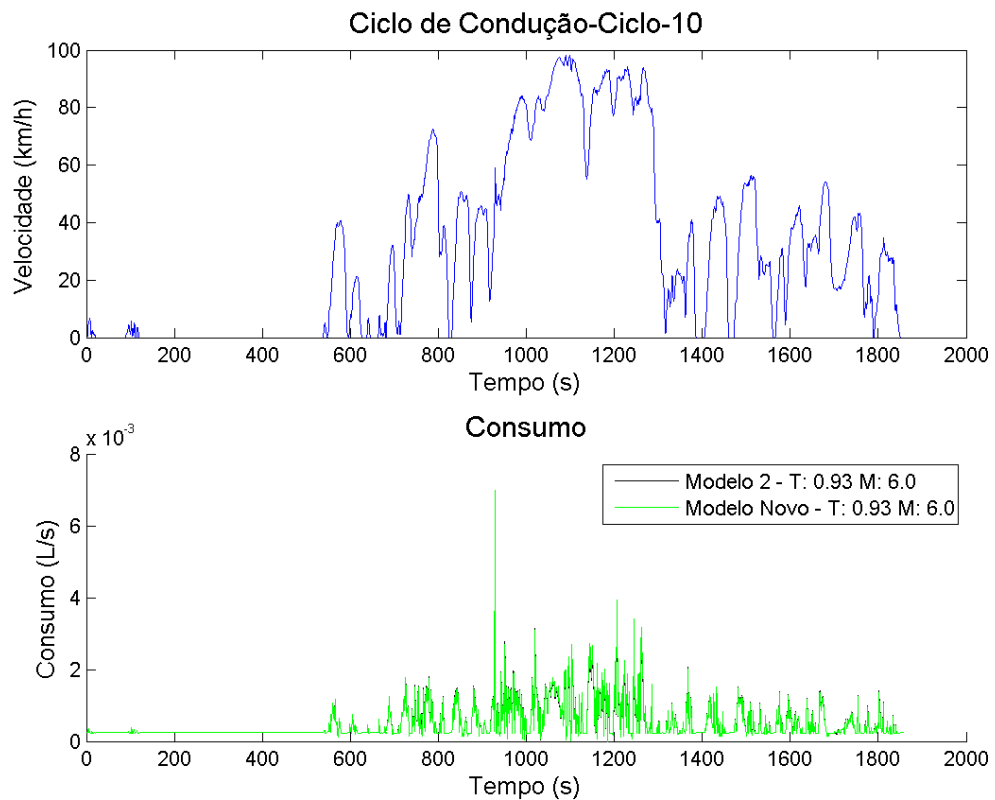


Figura 5.14: Apresentação do resultado final para ciclo 10

- Ciclo 14

Neste décimo quarto ciclo observa-se que, tal como no ciclo anterior, as diferenças são muito pequenas, sendo que em ambos estas estão perto do valor real de $6.9L/100km$.

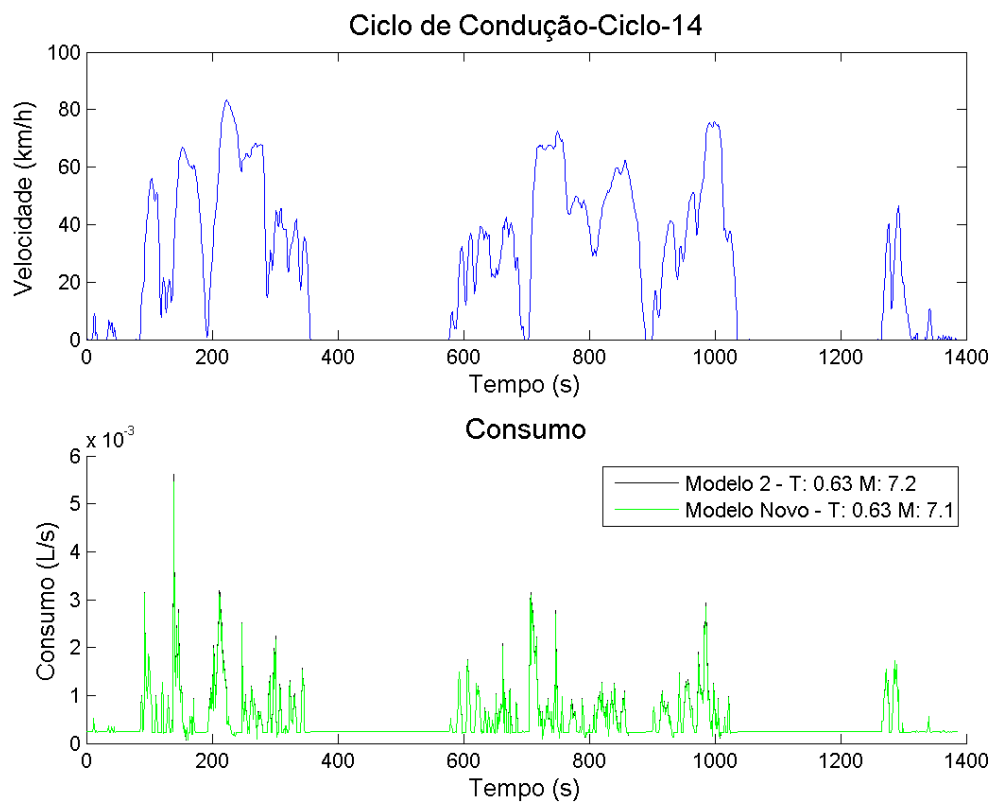


Figura 5.15: Apresentação do resultado final para ciclo 14

- Ciclo 35

Neste último ciclo é, novamente, possível constatar que o consumo é ligeiramente mais aproximado no caso do novo modelo em relação ao modelo anterior, dado que o consumo para este ciclo foi de $6.1L/100km$.

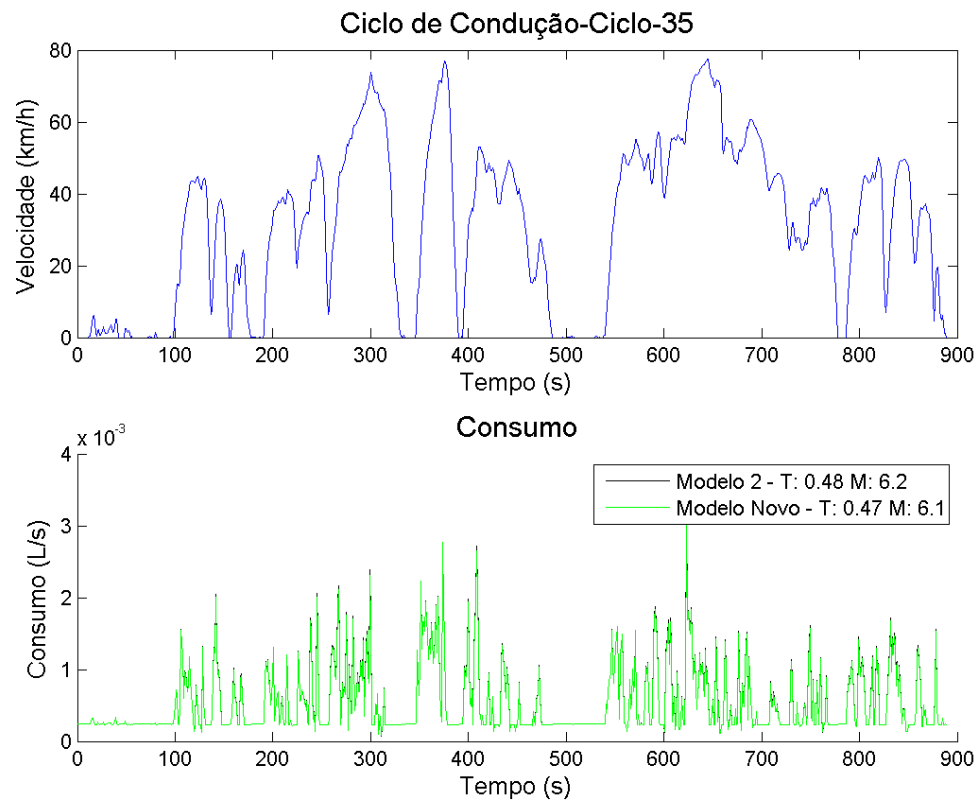


Figura 5.16: Apresentação do resultado final para ciclo 35

Ilustram-se, agora, os resultados da simulação do ciclo total, isto é, do ciclo acumulativo de todo o percurso (230.6km)

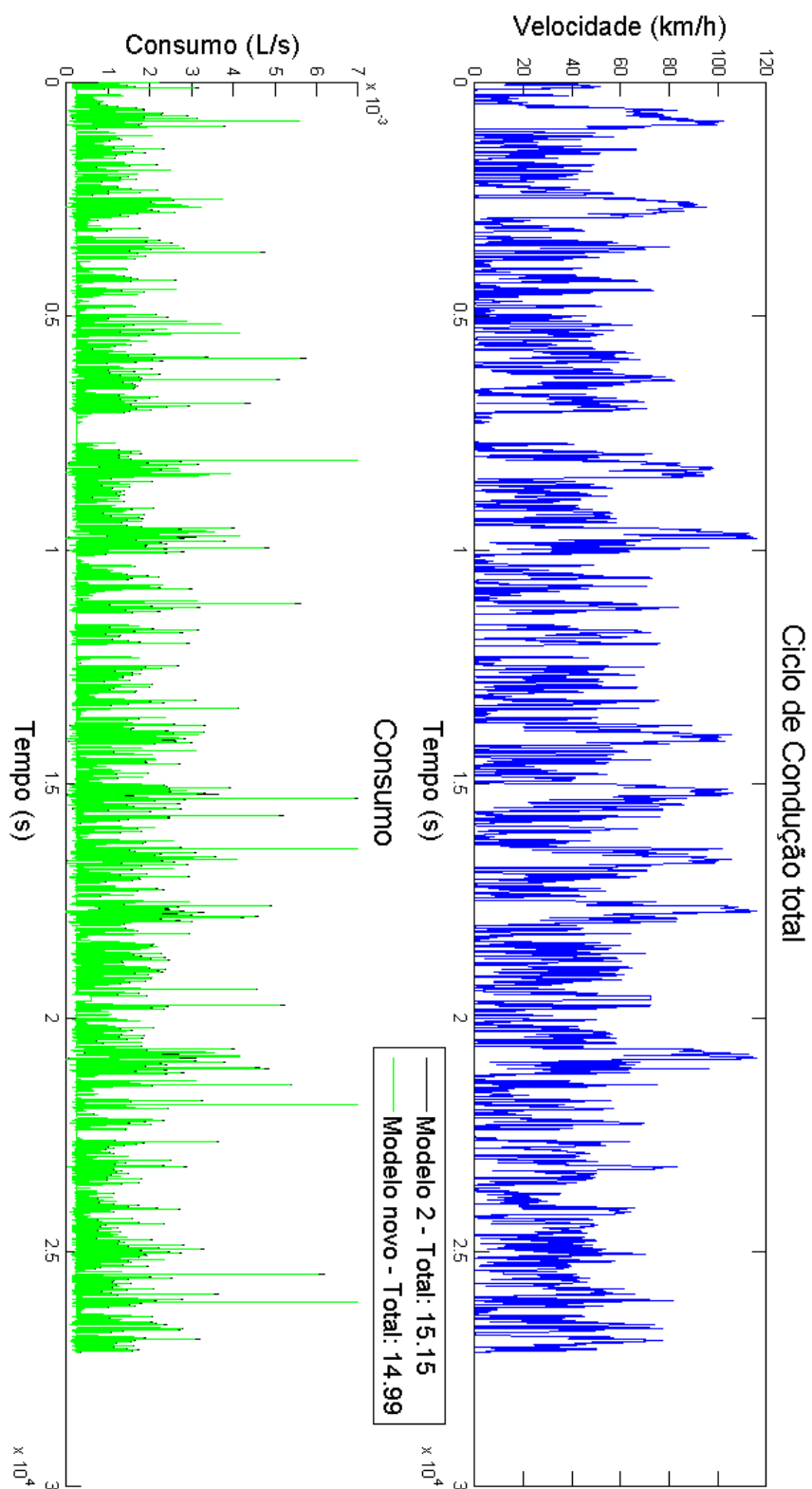


Figura 5.17: Apresentação do resultado final total

Tabela 5.4: Consumos e erros da estimativa de consumo

Modelo	Consumo Total (L)	Consumo médio (L/100km)	Erro absoluto (L/100km)	Erro relativo (%)
Modelo 2	15.15	6.5698	0.0698	1.07
Modelo novo	14.99	6.5004	4.3365×10^{-4}	0.01

Na figura e na tabela acima expostas é possível constatar que o consumo total é mais aproximado usando o novo algoritmo, sendo que, o erro relativo é muito menor no novo modelo. Assim, verifica-se que, numa viagem ainda mais longa, a estimativa do consumo do modelo 2 iria divergir mais do que para o novo modelo.

5.4 Conclusões

Tendo em consideração a implementação em Android e os respectivos resultados, exemplificativos, conclui-se que o novo modelo é ligeiramente melhor do que o modelo 2, já que, a longo prazo e percorrendo diferentes tipos de ciclos, consegue uma maior aproximação ao consumo real. Tal é conseguido aumentando a complexidade computacional. Porém, como verificado nos testes, o dispositivo Android foi capaz de funcionar com o novo modelo, podendo-se afirmar, deste modo, que o novo modelo apresenta vantagens relativamente ao modelo mais simples.

Após este estudo é, agora, importante avaliar como se comportam os dois algoritmos para diferentes veículos e, também, verificar de que forma se comporta o algoritmo em todos os instantes.

Capítulo 6

Conclusão

Esta dissertação, alicerçando-se em motivações diversas, em que se destacam preocupações ambientais e económicas associadas à procura de um maior conhecimento e saber no domínio das aplicações móveis, procurou sistematizar os modelos disponíveis para se estimar o consumo de combustível de um automóvel convencional, analisando com mais detalhe um desses modelos, por este estar mais em consonância com um dos objetivos traçados: possuir entradas passíveis de serem adquiridas por um dispositivo móvel. Tendo como pano de fundo o modelo criteriosamente escolhido desenvolveu-se um algoritmo, o qual se implementou num dispositivo Android, submetendo-o a vários testes, com vista a verificar-se os seus resultados.

6.1 Principais Resultados

Sumariamente, os resultados mais relevantes desta dissertação foram o desenvolvimento de um algoritmo de deteção de tipo de ciclo de condução e um algoritmo de estimativa de consumo de combustível que utiliza o primeiro no seu processo de estimativa. Estes algoritmos foram implementados num dispositivo móvel com sistema operativo Android, o que permitiu verificar se seria viável a sua aplicabilidade.

Em síntese, o primeiro algoritmo recorre ao conhecimento da velocidade média, ao tempo parado e à energia de aceleração média, nos últimos 50 segundos, para estimar com base num sistema de lógica difusa que tipo de ciclo o veículo se encontra a percorrer, sendo que estes ciclos são classificados em: *stop and go*, urbano, semi urbano, via rápida com tráfego, via rápida sem tráfego, auto estrada sem tráfego e auto estrada com tráfego. A saída do algoritmo define-se como um sinal limitado entre 0 e 100. Esta é dividida em vários escalões, cada um deles referente a um tipo de ciclo.

O segundo algoritmo, o de estimação de consumo de combustível, é baseado num modelo físico apresentado em [10] cujas entradas são a velocidade, a aceleração e a inclinação da estrada. Este possui um amplo leque de parâmetros que permitem distinguir tanto o ambiente envolvente, como as características do veículo. Assim, após se ter efetuado uma análise da sensibilidade do modelo em causa, verificando de que forma a variação das entradas afeta as saídas, e de se ter

analisado a influência da variação do período de amostragem, da introdução de ruído e do método utilizado para obter a aceleração como derivada da velocidade, conclui-se, pela sua robustez, que pode ser tomado como base para o desenvolvimento do segundo algoritmo presente nesta dissertação. Não obstante, modificou-se o algoritmo omitindo-se a inclinação da estrada, considerando-se apenas uma das eficiências apresentadas, sendo que esta deveria variar de acordo com o tipo de ciclo. Esta terá um valor inicial fixo, calibrado de acordo com o modelo presente em [10] e já analisado. Seguidamente terá, ainda, outro termo que depende da saída do algoritmo de detecção de tipo de ciclo de condução e de uma constante associada a cada um destes tipos.

Relativamente ao último resultado obtido, a implementação destes dois algoritmos no dispositivo com sistema operativo Android, no que se refere ao primeiro usaram-se as bibliotecas de lógica difusa apresentadas em [34, 35], que permitiram implementar o bloco de lógica difusa de forma simples e compacta. No que se refere ao segundo algoritmo, recorrendo às bibliotecas do Java e do Android, foi possível calcular, de forma simples, a estimativa de combustível. Sublinhe-se, ainda que, devido ao facto do algoritmo estar implementado num dispositivo móvel, se teve sempre em atenção o seu desempenho e, como tal, foram criados o máximo de processos paralelos durante a estimativa, de modo a que o cálculo fosse o mais rápido possível. Aliado a esta implementação, foi criado um sistema de aquisição, capaz de guardar todos os dados (velocidade, aceleração, posição global e data) para que, posteriormente, possam ser utilizados e analisados e, daí, poderem-se extrair conclusões em relação ao consumo.

Do atrás exposto, poder-se-á inferir que os resultados finais correspondem de forma positiva com o inicialmente proposto, principalmente, quando comparados com os resultados provenientes da plataforma Advisor, que, durante esta dissertação, serviu como ponto de referência para o cálculo do consumo de combustível.

6.2 Contribuições

A aplicação criada, tendo em consideração o modelo desenvolvido, permitirá obter uma estimativa do consumo instantâneo para um veículo de classe B, recorrendo, apenas, a um *smartphone* possuidor de GPS e capaz de processar os dados de forma satisfatória.

6.3 Trabalho futuro

Pensa-se que, no futuro, seria importante realizar ensaios, nos quais se procederia a uma aquisição em tempo real, para vários veículos. Deste modo, pretender-se-ia avaliar eficazmente o comportamento do estimador de combustível. Seria, ainda, interessante verificar de que forma a alteração de dispositivos influenciaria o tempo de processamento do estimador, bem como verificar o comportamento dos mesmos. Por fim, seria, igualmente, interessante estudar o impacto do estilo de condução na variação do consumo de combustível, acrescentando-se, portanto, mais um fator ao tipo de ciclo de condução.

Anexo A

Anexos do modelo simulado

A.1 Diagrama do algoritmo de identificação de tipo de ciclo

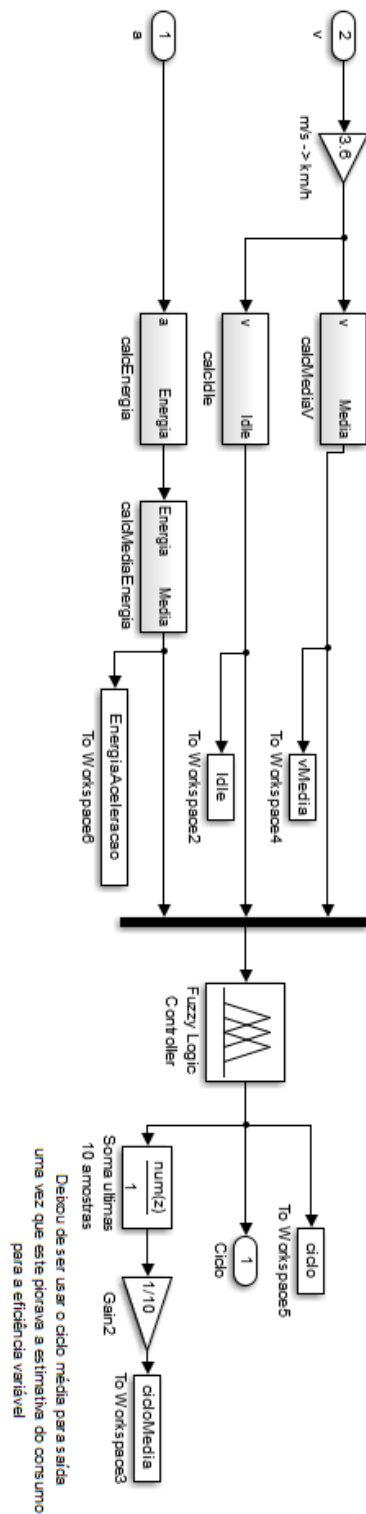
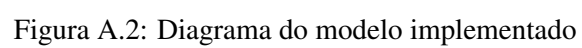


Figura A.1: Diagrama do modelo de identificação de tipo de ciclo

A.2 Diagrama do modelo implementado



Anexo B

Anexos do código implementado

B.1 Código da atividade principal

```
package moreiraeduardo.device;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.math.BigDecimal;
import java.util.Calendar;

import moreiraeduardo.estimacao.Car;
import moreiraeduardo.estimacao.CarEnvironment;
import moreiraeduardo.estimacao.Estimator;
import moreiraeduardo.estimacao.Position;
import moreiraeduardo.estimacao.VehicleState;

import moreiraeduardo.fce.R;

import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Environment;
```

```
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.MenuItem;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

public class Consumo extends Activity implements OnClickListener {

    private LocationManager mLocationManager;
    private LocationListener mLocationListener;

    private double longitude, latitude, speed, altitude, acceleration, ciclo;
    private String longitudeString, latitudeString, speedString,
    altitudeString, accelerationString;

    private float totalDist, lastSpeed = 0;

    // File Variables
    File dir;
    File myFile = null;
    FileOutputStream fOut = null;
    OutputStreamWriter myOutWriter = null;
    String FileName;

    // Miscellaneous Variables
    String timestamp;

    private boolean state = false;
    private boolean gpsAvailable = false;

    private TextView textLati, textLongi, textVel, textDist, textDebug, textNS,
    textEO, textTotalConsume, textConsume, textUniVel, textUniCons;

    private Button bStart;
```

```
private Car mCar;
private CarEnvironment mCarEnvironment;
private Estimator mEstimator;
private VehicleState mVehicleState;
private Position mPosition;
private double consumoI;

// Methods Variables
boolean lastWrite = false;
boolean shutDown = false;
RadioGroup metodoA;
Thread t, t_cycle;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Preparing Content View
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.consumo);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON,
        WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

    // Data visualization
    textConsume = (TextView) findViewById(R.id.consume);
    textConsume.setText("0");
    textUniCons = (TextView) findViewById(R.id.unidadesConsume);
    textUniCons.setText("L/100km");

    textVel = (TextView) findViewById(R.id.valorVelocidade);
    textUniVel = (TextView) findViewById(R.id.unidadesVelocidade);
    textUniVel.setText("km/h");

    textLati = (TextView) findViewById(R.id.valorLatitude);
    textLongi = (TextView) findViewById(R.id.valorLongitude);
    textNS = (TextView) findViewById(R.id.norteSul);
    textEO = (TextView) findViewById(R.id.esteOeste);

    textDist = (TextView) findViewById(R.id.distanciaV);
```

```
textTotalConsume = (TextView) findViewById(R.id.totalConsume);
textTotalConsume.setText("0");

// Status Text
textDebug = (TextView) findViewById(R.id.textStatus);

// Button
bStart = (Button) findViewById(R.id.bStartStop);
bStart.setOnClickListener(this);

// Declaration of the directory to save
dir = new File(Environment.getExternalStorageDirectory()
+ "/Fuel Estimator");
dir.mkdirs();

mCarEnvironment = new CarEnvironment((float) 1.225);
mCar = new Car((float) 0.29, (float) 2.487, (float) 0.29, (float) 0.335,
(float) 0.23, (float) 36196, (float) 1215, (float) 50);

mEstimator = new Estimator(mCar, mCarEnvironment);
mVehicleState = new VehicleState();
mPosition = new Position(0, 0, 0);

// GPS Localization Declaration
mLocationManager = (LocationManager) this
.getSystemService(Context.LOCATION_SERVICE);

mLocationListener = new LocationListener() {
public void onLocationChanged(Location location) {

if (location != null) {

longitude = location.getLongitude();
latitude = location.getLatitude();
altitude = location.getAltitude();

speed = location.getSpeed();

if (speed - lastSpeed > 2) {
speed = lastSpeed;
```

```
} else if (speed - lastSpeed < -4) {
    speed = lastSpeed;
}

acceleration = speed - lastSpeed;
lastSpeed = (float) speed;

mPosition.setPosition((float) longitude, (float) latitude,
    (float) altitude);

longitudeString = roundToString((float) longitude, 6);
if (longitude < 0) {
    longitude = -longitude;
    textEO.setText("W");
} else {
    textEO.setText("E");
}

String aux = Location.convert(longitude,
    Location.FORMAT_SECONDS);
String[] partes = aux.split(":");
String[] sec1 = partes[2].split(",");
textLongi.setText(partes[0] + "°" + partes[1] + "' "
    + sec1[0] + "." + sec1[1].charAt(0) + "'");

latitudeString = roundToString((float) latitude, 6);
if (latitude < 0) {
    latitude = -latitude;
    textNS.setText("S");
} else {
    textNS.setText("N");
}

aux = Location.convert(latitude, Location.FORMAT_SECONDS);
String[] partes2 = aux.split(":");
String[] sec2 = partes2[2].split(",");
textLati.setText(partes2[0] + "°" + partes2[1] + "' "
    + sec2[0] + "." + sec2[1].charAt(0) + "'");
```

```
altitudeString = roundToString((float) altitude, 2);

speedString = roundToString((float) speed, 3);

totalDist = (float) (totalDist + speed / 1000);

String speedkmh = roundToString((float) (speed * 3.6), 0);
textVel.setText(speedkmh);

accelerationString = roundToString((float) acceleration, 3);

mVehicleState.setData(mPosition, (float) speed,
    (float) acceleration, 0);

if (ciclo > 90) {
    textDebug.setText("Auto Estrada Sem Tráfego");
} else if (ciclo > 75) {
    textDebug.setText("Auto Estrada Com Tráfego");
} else if (ciclo > 60) {
    textDebug.setText("Via Rápida Com Tráfego");
} else if (ciclo > 40) {
    textDebug.setText("Via Rápida Sem Tráfego");
} else if (ciclo > 25) {
    textDebug.setText("Semi Urbano");
} else if (ciclo > 10) {
    textDebug.setText("Urbano");
} else {
    textDebug.setText("Stop and Go");
}

// Descomentar para guardar tipo de ciclo no lugar da altitude
//altitudeString = roundToString((float) ciclo, 2);

if (speed < 5 || consumoI < 0 ) {
    textConsume.setText("--");
} else {
```



```
textConsume.setText(roundToString((float) consumoI, 1));
}

textDist.setText(roundToString(totalDist, 2) + "km");

String totCons = roundToString(
mEstimator.getTotalConsume(), 3) + "L";
textTotalConsume.setText(totCons);
gpsAvailable = true;
}

}

public void onStatusChanged(String provider, int status,
Bundle extras) {

}

public void onProviderEnabled(String provider) {

}

public void onProviderDisabled(String provider) {
gpsAvailable = false;

}
};

mLocationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
0, 0, mLocationListener);

t = new Thread() {
public void run() {

long currentTime, lastTime;
currentTime = System.nanoTime();
lastTime = currentTime;
t.setPriority(MAX_PRIORITY);
while (true) {
```

```
currentTime = System.nanoTime();

if (currentTime - lastTime >= 1000000000) {
    lastTime = currentTime;

    if (state) {
        writeDataGroup();
    }
    if (lastWrite) {
        try {
            myOutWriter.close();
            fOut.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        lastWrite = false;
    }
    if (shutDown) {
        try {
            t.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

}

};
t.start();

t_cycle = new Thread() {
    public void run() {

        long currentTime, lastTime;
        currentTime = System.nanoTime();
        lastTime = currentTime;

        while (true) {
```

```
currentTime = System.nanoTime();

if (currentTime - lastTime >= 1000000000) {
    lastTime = currentTime;

    ciclo = mVehicleState.calcCycle();
    mCar.setVariableEfficiency(ciclo);

    consumoI = mEstimator
        .calculateConsume(mVehicleState);

    consumoI = 100 * (consumoI / (speed / 1000));
}
if (shutDown) {
    try {
        t_cycle.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

}

};
t_cycle.start();
}

@Override
public boolean onCreateOptionsMenu(android.view.Menu menu) {

    getMenuInflater().inflate(R.menu.menu_consumo, menu);

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
```

```
case R.id.menuUnidades:
    Toast.makeText(Consumo.this, "Menu Unidades", Toast.LENGTH_SHORT)
        .show();
    break;
case R.id.menuDefinicoes:
    Toast.makeText(Consumo.this, "Menu Definições", Toast.LENGTH_SHORT)
        .show();
    break;
case R.id.menuVeiculo:
    Toast.makeText(Consumo.this, "Menu Veículo", Toast.LENGTH_SHORT)
        .show();
    break;
}
return super.onOptionsItemSelected(item);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    shutDown = true;
    locationManager.removeUpdates(mLocationListener);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.bStartStop) {
        if (gpsAvailable) {

            if (state) {
                state = false;
                lastWrite = true;
                textDebug.setText("Status: Em Espera");
                textDebug.setTextColor(Color.BLACK);
                Toast.makeText(Consumo.this, "Terminou aquisição",
                    Toast.LENGTH_SHORT).show();
            } else {
                try {
                    createFile();
                } catch (IOException e1) {
                    e1.printStackTrace();
                }
            }
        }
    }
}
```

```
}

state = true;
textDebug.setText("Status: Activo");
textDebug.setTextColor(Color.GREEN);
totalDist = 0;
mEstimator.setTotalConsume(0);
Toast.makeText(Consumo.this, "Começou aquisição",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(Consumo.this, "GPS não disponível",
Toast.LENGTH_SHORT).show();
}
}
}

protected boolean createFile() throws IOException {

Calendar today = Calendar.getInstance();

FileName = String.valueOf(today.get(Calendar.YEAR)) + "-"
+ String.valueOf(today.get(Calendar.MONTH) + 1) + "-"
+ String.valueOf(today.get(Calendar.DAY_OF_MONTH)) + "_"
+ getTimeStamp("-") + ".txt";

myFile = new File(dir, FileName);

try {
fOut = new FileOutputStream(myFile);
myOutWriter = new OutputStreamWriter(fOut);
myOutWriter
.write("Time\tLongitude\tLatitude\tAltitude\tSpeed\tAcceleration\r\n");
return true;
} catch (FileNotFoundException e) {
e.printStackTrace();
return false;
}
}
```

```
protected String getTimeStamp(String separator) {
    Calendar now = Calendar.getInstance();
    return String.valueOf(now.get(Calendar.HOUR_OF_DAY)) + separator
    + String.valueOf(now.get(Calendar.MINUTE)) + separator
    + String.valueOf(now.get(Calendar.SECOND));
}

protected void writeDataGroup() {

    timestamp = getTimeStamp(":");

    try {
        if (isExternalStorageWritable()) {
            myOutWriter.append(timestamp + "\t" + longitudeString + "\t"
            + latitudeString + "\t" + altitudeString + "\t"
            + speedString + "\t" + accelerationString + "\r\n");
        }
    } catch (IOException e) {
        Log.e(Posicao.class.getName(), "File write failed: " + e.toString());
    }
}

public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

public static String roundToString(float d, int decimalPlace) {
    BigDecimal bd = new BigDecimal(Float.toString(d));
    bd = bd.setScale(decimalPlace, BigDecimal.ROUND_HALF_UP);
    return bd.toString();
}
}
```

B.2 Código das classes dos estimador

B.2.1 Classe Car

```
package moreiraeduardo.estimacao;

public class Car {
    private float efficiency;
    private float frontalArea;
    private float rollingLoss;
    private float airDrag;
    private float idleConsume;
    private float heatEquiDiesel;
    private float mass;
    private float movingMass;
    private float fixedEfficiency;

    private float autoST;
    private float autoCT;
    private float viaRapCT;
    private float viaRapST;
    private float semiUrb;
    private float urb;
    private float stopGo;

    public Car(float efficiency, float frontalArea, float rollingLoss,
        float airDrag, float idleConsume, float heatEquiDiesel, float mass,
        float movingMass) {

        this.efficiency = efficiency;
        this.frontalArea = frontalArea;
        this.rollingLoss = rollingLoss;
        this.airDrag = airDrag;
        this.idleConsume = idleConsume;
        this.heatEquiDiesel = heatEquiDiesel;
        this.mass = mass;
        this.movingMass = movingMass;
        this.fixedEfficiency = efficiency;
    }
}
```

```
    autoST=1/6000;
    autoCT=1/1500;
    viaRapCT=1/8000;
    viaRapST=-1/3000;
    semiUrb=1/6000;
    urb=1/2000;
    stopGo=1/1000;

}

public float getEfficiency() {
    return efficiency;
}

public void setEfficiency(float efficiency) {
    this.efficiency = efficiency;
}

public void setVariableEfficiency(double cycle) {

    float constant;

    if (cycle > 90){
        constant = autoST;
    }else if (cycle > 75){
        constant = autoCT;
    }else if (cycle > 60){
        constant = viaRapCT;
    }else if (cycle > 40){
        constant = viaRapST;
    }else if (cycle > 25){
        constant = semiUrb;
    }else if (cycle > 10){
        constant = urb;
    }else{
        constant = stopGo;
    }

    this.efficiency = (float) (fixedEfficiency + constant * cycle);
}
```



```
}

public float getFrontalArea() {
    return frontalArea;
}

public void setFrontalArea(float frontalArea) {
    this.frontalArea = frontalArea;
}

public float getRollingLoss() {
    return rollingLoss;
}

public void setRollingLoss(float rollingLoss) {
    this.rollingLoss = rollingLoss;
}

public float getAirDrag() {
    return airDrag;
}

public void setAirDrag(float airDrag) {
    this.airDrag = airDrag;
}

public float getIdleConsume() {
    return idleConsume;
}

public void setIdleConsume(float idleConsume) {
    this.idleConsume = idleConsume;
}

public float getHeatEquiDiesel() {
    return heatEquiDiesel;
}

public void setHeatEquiDiesel(float heatEquiDiesel) {
    this.heatEquiDiesel = heatEquiDiesel;
}
```

```
}

public float getMass() {
    return mass;
}

public void setMass(float mass) {
    this.mass = mass;
}

public float getMovingMass() {
    return movingMass;
}

public void setMovingMass(float movingMass) {
    this.movingMass = movingMass;
}

}
```

B.2.2 Classe CarEnvironment

```
package moreiraeduardo.estimacao;

public class CarEnvironment {
    private float gravity;
    private float airDensity;

    public CarEnvironment(float airDensity) {

        this.gravity = (float) 9.81;
        this.airDensity = airDensity;
    }

    public float getGravity() {
        return gravity;
    }

    public void setGravity(float gravity) {
        this.gravity = gravity;
    }
}
```

```
public float getAirDensity() {  
    return airDensity;  
}  
  
public void setAirDensity(float airDensity) {  
    this.airDensity = airDensity;  
}  
  
}
```

B.2.3 Classe VehicleState

```
package moreiraeduardo.estimacao;  
  
import android.os.Environment;  
import net.sourceforge.jFuzzyLogic.FIS;  
  
public class VehicleState {  
    private Position position;  
    private float speed;  
    private float acceleration;  
    private float roadSlope;  
    private float lastAce, lastLastAce;  
    private float[] lastSpeed;  
    private float[] lastEnergyAce;  
    private int[] idleTime;  
    private int i;  
    private float AvgSpeed;  
    private float AvgEnergyAce;  
    private float sumIdleTime;  
    private double cycle;  
  
    private FIS fis;  
  
    public void setData(Position position, float speed, float acceleration, float roadSlope) {  
        this.speed = speed;  
        this.acceleration = acceleration;  
        this.position = position;  
        this.roadSlope = roadSlope;  
    }  
}
```

```
lastSpeed[i] = speed;
lastEnergyAce[i] = lastAce*lastAce - lastLastAce*this.acceleration;
lastLastAce = lastAce;
lastAce = this.acceleration;

if ( (speed*3.6) < 5){
idleTime[i] = 1;
}else{
idleTime[i] = 0;
}

AvgSpeed = 0;
AvgEnergyAce = 0;
sumIdleTime = 0;

for(int k=0 ;k < 50;k++){
AvgSpeed += lastSpeed[k];
AvgEnergyAce += lastEnergyAce[k];
sumIdleTime += idleTime[k];
}

int n=lastSpeed.length;

if(i == 0){
AvgSpeed = (float) (3.6*((0.8*AvgSpeed/n)+(0.2*lastSpeed[49])));
AvgEnergyAce = (float) (0.8*AvgEnergyAce/n+0.2*lastEnergyAce[49]);
sumIdleTime = sumIdleTime / n * 100;
}
else{
AvgSpeed = (float) (3.6*((0.8*AvgSpeed/n)+(0.2*lastSpeed[i-1])));
AvgEnergyAce = (float) (0.8*AvgEnergyAce/n+0.2*lastEnergyAce[i-1]);
sumIdleTime = sumIdleTime / n * 100;
}
```

```
if (i < 49){
    i++;
}else{
    i = 0;
}
}

public double calcCycle(){
    this.cycle = 0;
    // Set inputs
    //      fis.setVariable("velocidadeMedia", AvgSpeed);
    //      fis.setVariable("energiaAceleracao", AvgEnergyAce);
    //      fis.setVariable("idle", sumIdleTime);
    //
    //      // Evaluate
    //      fis.evaluate();
    //
    //      // Show output variable's chart
    //      this.cycle = fis.getVariable("ciclo").getValue();

    return this.cycle;
}

public VehicleState() {
    lastSpeed = new float[50];
    lastEnergyAce = new float[50];
    idleTime = new int[50];
    i=0;

    String fileName = Environment.getExternalStorageDirectory()
+ "/Fuel Estimator/determinaCiclo.fcl";
    fis = FIS.load(fileName,true);

    // Error while loading?
    if( fis == null ) {
        System.err.println("Can't load file: '" + fileName + "'");
    }
}
```

```
}

public float getAvgSpeed(){
return AvgSpeed;
}

public float getEnergyAce(){
return AvgEnergyAce;
}

public float getIdleTime(){
return sumIdleTime;
}

public Position getPosition() {
return position;
}

public float getSpeed() {
return speed;
}

public float getAcceleration() {
return acceleration;
}

public float getRoadSlope() {
return roadSlope;
}

}
```

B.2.4 Classe Position

```
package moreiraeduardo.estimacao;
```

```
public class Position {
    private float longitude;
    private float latitude;
    private float altitude;

    public Position(float longitude, float latitude, float altitude) {

        this.longitude = longitude;
        this.latitude = latitude;
        this.altitude = altitude;
    }

    public void setPosition(float longitude, float latitude, float altitude) {

        this.longitude = longitude;
        this.latitude = latitude;
        this.altitude = altitude;
    }

    public float getLongitude() {
        return longitude;
    }

    public float getLatitude() {
        return latitude;
    }

    public float getAltitude() {
        return altitude;
    }

}
```

B.2.5 Classe Estimator

```
package moreiraeduardo.estimacao;

public class Estimator {
    private Car car;
    private VehicleState vehicleState;
```

```
private CarEnvironment carEnvironment;
private float totalConsume;

public Estimator(Car car, CarEnvironment carEnvironment) {
    this.car = car;
    this.carEnvironment = carEnvironment;
    totalConsume = 0;
}

public Car getCar() {
    return car;
}

public void setCar(Car car) {
    this.car = car;
}

public CarEnvironment getCarEnvironment() {
    return carEnvironment;
}

public void setCarEnvironment(CarEnvironment carEnvironment) {
    this.carEnvironment = carEnvironment;
}

public float calculateConsume(VehicleState vehState) {
    this.vehicleState = vehState;
    float newConsume;
    float term1, term2, term3;

    term1 = (car.getMass() + car.getMovingMass())
    * vehicleState.getAcceleration();

    term2 = car.getAirDrag() * car.getFrontalArea()
    * carEnvironment.getAirDensity() * vehicleState.getSpeed()
    * vehicleState.getSpeed() / 2;

    term3 = (float) (car.getMass() * carEnvironment.getGravity() * (car
    .getRollingLoss() * Math.cos(vehicleState.getRoadSlope()) + Math
```



```
.sin(vehicleState.getRoadSlope()))));

float fw = term1 + term2 + term3;

if (fw > 1) {
    newConsume = ((term1 + term2) * vehicleState.getSpeed() + term3)
    * 1 / (car.getEfficiency() * car.getHeatEquiDiesel());
} else {
    newConsume = 0;
}

newConsume += car.getIdleConsume();

newConsume = newConsume/1000;

if (newConsume > 0.007){
    newConsume = (float) 0.007;
}

totalConsume += newConsume;
return newConsume;
}

public float getTotalConsume() {
    return totalConsume;
}

public void setTotalConsume(float totalConsume) {
    this.totalConsume = totalConsume;
}
}
```


Referências

- [1] I. Taymaz e M. Benli. Emissions and fuel economy for a hybrid vehicle. *Fuel*, 115:812–817, 2014. Export Date: 27 January 2014 Source: Scopus.
- [2] Georgios Fontaras, Panayotis Pistikopoulos, e Zissis Samaras. Experimental evaluation of hybrid vehicle fuel economy and pollutant emissions over real-world simulation driving cycles. *Atmospheric Environment*, 42(18):4023–4035, 2008.
- [3] Phil Goodwin, Joyce Dargay, e Mark Hanly. Elasticities of road traffic and fuel consumption with respect to price and income: A review. *Transport Reviews*, 24(3):275–292, 2004.
- [4] Hsiao Kuo-Lun. Android smartphone adoption and intention to pay for mobile internet: Perspectives from software, hardware, design, and value. *Library Hi Tech*, 31(2):216–235, 2013.
- [5] J. Karikoski e T. Soikkeli. Contextual usage patterns in smartphone communication services. *Personal and Ubiquitous Computing*, 17(3):491–502, 2013.
- [6] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, e M. C. Gonzalez. Safe driving using mobile phones. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1462–1468, 2012.
- [7] B. Beusen e T. Denys. *Long-term effect of eco-driving education on fuel consumption using an on-board logging device*, volume 101 de *Wit Transactions on the Built Environment*, páginas 395–403. Wit Press, Southampton, 2008.
- [8] D. A. Johnson e M. M. Trivedi. Driving style recognition using a smartphone as a sensor platform. Em *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, páginas 1609–1615.
- [9] Américo Vicente Teixeira Leite. *Estimação de estados, parâmetros e velocidade do motor de indução trifásico com metodologias de identificação em tempo real*. Thesis, Faculdade de Engenharia da Universidade do Porto, 2004.
- [10] Daniel Macias. *Estimation of Fuel Consumption for Real Time Implementation*. Thesis, KTH Electrical Engineering, 2012.
- [11] Imen Zaabar. *Effect of pavement condition on vehicle operating costs including fuel consumption, vehicle durability and damage to transported goods*. Thesis, Michigan State University, 2010.
- [12] R Giannelli, EK Nam, Kent Helmer, Theodore Younglove, George Scora, e M Barth. Heavy-duty diesel vehicle fuel consumption modeling based on road load and power train parameters. 2005.

- [13] Takashi Oguchi, Masahiko Katakura, e Masaaki Taniguchi. Available concepts of energy reduction measures against road vehicular traffic. Em *Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport Systems*.
- [14] C.J. Bester. *Fuel Consumption of Highway Traffic*. Thesis, 1981.
- [15] DC Biggs. Estimating fuel consumption of light to heavy vehicles. Report, 1987.
- [16] D. C. Australian Road Research Board Biggs. *ARFCOM : models for estimating light to heavy vehicle fuel consumption*. Australian Road Research Board, Vermont South, Vic., 1988.
- [17] K Post, JH Kent, J Tomlin, e N Carruthers. Fuel consumption and emission modelling by power demand and a comparison with other models. *Transportation Research Part A: General*, 18(3):191–213, 1984.
- [18] I. Greenwood, R. Dunn, e R. Raine. Estimating the effects of traffic congestion on fuel consumption and vehicle emissions based on acceleration noise. *Journal of Transportation Engineering*, 133(2):96–104, 2007.
- [19] Hesham Rakha, Kyoungcho Ahn, Kevin Moran, Bart Saerens, e Eric Van den Bulck. Simple comprehensive fuel consumption and co2 emissions model based on instantaneous vehicle power. Em *90th Transportation Research Board Annual Meeting*.
- [20] Earl C Klaubert. Highway effects on vehicle performance. Report, 2001.
- [21] Kyoungcho Ahn, Hesham Rakha, Antonio Trani, e Michel Van Aerde. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *Journal of Transportation Engineering*, 128(2):182–190, 2002.
- [22] M. Alsabaan, K. Naik, e T. Khalifa. Optimization of fuel cost and emissions using v2v communications. *Intelligent Transportation Systems, IEEE Transactions on*, 14(3):1449–1461, 2013.
- [23] Jorge Miguel Aroso Dias Ferreira. Sistema de aquisição de dados para um veículo eléctrico. Thesis, Faculdade de Engenharia da Universidade do Porto, 2010.
- [24] C. Coopmans e Chen Yang-Quan. A general-purpose low-cost compact spatial-temporal data logger and its applications. Em *AUTOTESTCON, 2008 IEEE*, páginas 64–68.
- [25] B. R. Barkstrom. When is it sensible not to use xml? *Earth Science Informatics*, 4(1):45–53, 2011.
- [26] G. Del Vescovo, M. Paschero, A. Rizzi, e F. M. F. Mascioli. An open software system for signal routing and processing in hybrid vehicles. Em *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, páginas 1702–1707.
- [27] Inc Alliance for Sustainable Energy, LLC. Advisor documentation, 2013. URL: <http://adv-vehicle-sim.sourceforge.net/>.
- [28] L. P. Huelsman e P. E. Allen. *Introduction to the Theory and Design of Active Filters*. McGraw-Hill Book, New York, 1980. eng.
- [29] Heitor Pina. *Métodos numéricos*. McGraw-Hill, Lisboa, 1995. por.

- [30] R. Araujo, A. Igreja, R. de Castro, e R. E. Araujo. Driving coach: A smartphone application to evaluate driving efficient patterns. Em *Intelligent Vehicles Symposium (IV)*, 2012 IEEE, páginas 1005–1010.
- [31] P. Maragos, J. F. Kaiser, e T. F. Quatieri. On amplitude and frequency demodulation using energy operators. *Signal Processing, IEEE Transactions on*, 41(4):1532–1550, 1993.
- [32] Y. L. Murphey, Park Jungme, Chen Zhihang, M. L. Kuang, M. A. Masrur, e A. M. Phillips. Intelligent hybrid vehicle power control - part i: Machine learning of optimal vehicle power. *Vehicular Technology, IEEE Transactions on*, 61(8):3519–3530, 2012.
- [33] S. Sakhi, A. Khetat, M. Hanks, M. Belhocine, A. Elouardi, e S. Bouaziz. An embedded data logger based vehicle trajectory monitoring system. Em *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on*, páginas 11–16.
- [34] Pablo Cingolani e Jesus Alcala-Fdez. jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation. Em *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, páginas 1–8. IEEE, 2012.
- [35] Pablo Cingolani e Jesús Alcalá-Fdez. jfuzzylogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming.
- [36] Carfolio. 2013 alfa romeo mito twinair specifications, information, data, photos 357720, 2013. URL: <http://www.carfolio.com/specifications/models/car/?car=357720>.